АЛЕКСЕЙ ПЕТЮШКИН

HTML

Экспресс курс



- Характеристика современных браузеров
- Форматирование текста и работа со шрифтами
- Работа с графикой
- Использование аудио- и видеофрагментов
- Применение визуальных динамических фильтров

Алексей Петюшкин

HTML
экспресс-курс

Санкт-Петербург

«БХВ-Петербург»

2003

УДК 681.3.06
ББК 32.973.26-018
П29

Петюшкин А. В.

П29 HTML. Экспресс-курс. — СПб.: БХВ-Петербург, 2003. — 256 с.: ил.

ISBN 5-94157-317-0

Книга представляет собой руководство по изучению основных правил языка гипертекстовой разметки HTML, необходимого практически каждому пользователю Всемирной сети, уже имеющему собственную Web-страницу в Интернете или только еще желающему ее создать. Дано общее представление о языке HTML, об интерактивных технологиях, реализующих электронные документы с динамически изменяющимся содержанием, раскрыта структура и описаны конструктивные элементы языка — теги и их параметры, приведены примеры различных способов форматирования текста электронных документов и соответствующие иллюстрации. Уделено внимание правилам использования графики и звука при создании Web-страниц, изложены принципы формирования карт-изображений, фреймов и пользовательских форм.

Для широкого круга пользователей Интернета

УДК 681.3.06
ББК 32.973.26-018

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Римма Смоляк*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 19.04.03.
Формат $70\times100^{1}/_{16}$. Печать офсетная. Усл. печ. л. 20,64.
Тираж 5000 экз. Заказ № 863
"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-317-0

© Петюшкин А. В., 2003
© Оформление, издательство "БХВ-Петербург", 2003

Содержание

Предисловие

Содержание книги

Книга "HTML. Экспресс-курс" посвящена описанию и разъяснению основ языка гипертекстовой разметки HTML, который в настоящее время является основой разработки электронных документов для Интернета.

Вследствие постоянного развития HTML-технологии, возникновения новых свойств и параметров, а также появления новых версий популярных браузеров, возможности HTML приобретают все большее значение в ходе создания Web-сайтов различной сложности и тематики. Знание основ языка HTML становится обязательным и неотъемлемым атрибутом многих специалистов в области интернет-технологий, превращается в систему знаний, необходимых практически каждому пользователю Всемирной сети.

Предлагаемая книга ориентирована на широкий круг читателей, желающих научиться создавать привлекательные Web-сайты с помощью языка гипертекстовой разметки HTML, а также интерактивных технологий Dynamic HTML и CSS. Материал, представленный в книге, изложен простым, понятным языком, содержит множество пояснительных иллюстраций и практических примеров с пошаговыми инструкциями и раскрывает все стороны создания электронных HTML-документов -- от форматирования текста и стилевых шаблонов CSS до применения карт-изображений и пользовательских форм.

Прочитав эту книгу, вы сможете в кратчайшие сроки овладеть популярным языком разметки HTML и создавать на его основе собственные Web-сайты и интерактивные документы различной сложности и тематической направленности.

Книга разделена на две части. "Часть I. HTML: первые шаги" раскрывает сущность языка разметки HTML и описывает следующие темы:

- применение браузеров;
- структура HTML-документа;
- форматирование текста и работа со шрифтами;
- структурное форматирование и блоки информационных данных;
- нумерованные и маркированные списки, списки определений;

- создание простых и вложенных таблиц, оптимизация табличных данных, использование таблицы в качестве структурной основы HTML-документа;
- графические форматы и типы файлов, используемые в Интернете;
- внедрение графики в Web-страницу;
- использование карт-изображений (Imagemap);
- разработка фреймов;
- внедрение звука в Web-страницу, обзор основных аудиоформатов, используемых в Интернете.

"Часть II. Создание интерактивных HTML-документов" знакомит читателя с основами разработки интерактивных электронных документов и освещает следующие темы:

- разработка пользовательских форм;
- основы применения каскадных таблиц стилей CSS;
- использование Dynamic HTML;
- позиционирование и визуальные динамические фильтры.

Книга "HTML. Экспресс-курс" -- это путь к быстрому и эффективному освоению навыков создания Web-документов с помощью языка HTML без утомительного изучения сотен страниц с техническими характеристиками и бесконечными листингами непонятного кода. В ней содержится только самая нужная и полезная для начинающего пользователя информация, способная помочь ему в дальнейшем закрепить полученные в ходе прочтения этой книги знания на практике.

Об авторе

Петюшкин Алексей Валерьевич закончил Санкт-Петербургский гуманитарный университет профсоюзов по специальности "Рекламное дело". Работает в интернет-индустрии свыше 7 лет, является автором более 100 аналитических и научных статей, посвященных вопросам информационных технологий, психологии рекламы, электронной коммерции, интернет-маркетинга, пользовательских интерфейсов и пр.

Статьи автора опубликованы во многих известных оффлайновых и сетевых изданиях, а также на серверах, посвященных интернет-технологиям. Среди них такие, как "CHIP-Россия", "ComputerPrice", "Мир Интернет", "Компьютерная Россия", "Магия ПК", "Новое Знание", "Маркетер", "Banner Index", "Raskrutka.Net", "eManual", "Компьютерная газета", "Санкт-Петербургские ведомости" и др.

Автор является сертифицированным специалистом по дисциплинам "Нейролингвистическое программирование (НЛП)", "Интернет-маркетинг", "Язык разметки HTML".

Принимал участие и проводил консультации по разработке и продвижению в более тридцати интернет-проектов и электронных презентаций, в числе которых такие, как "GetInfo.Ru", "URSA-Россия", "MS-Soft", "TravellingRussia", "ТехноподиуМ", "Guides of St.-Petersburg", "Griffin Securities", "Система документооборота Комитета экономики и инвестиций Правительства Ленинградской области", "Информационно-справочная система TopPlan", "Интранет-система Ford Motor Russia" и пр.

В апреле 2002 года издательство "БХВ-Петербург" выпустило книгу автора "Основы баннерной рекламы". Книга была рекомендована профессорско-преподавательским составом кафедры экономики и управления Санкт-Петербургского института международных экономических отношений (ИМЭО) в качестве учебного пособия по дисциплине "Менеджер по маркетингу в Интернете".

Свои комментарии, замечания, а также предложения по книге "HTML Экспресс-курс" можно оставить на авторском сайте по адресу **http://www.alpet.spb.ru.**

Часть I

HTML — первые шаги

Глава 1

Ваш первый HTML-документ

Что такое HTML

Всемирная паутина (World Wide Web — WWW) состоит из множества связанных между собой электронных документов, представляющих кладезь информационных данных, описанных с помощью специальных технологических правил. Эти правила составляются на языке гипертекстовой разметки HTML (HyperText Markup Language).

Можно с уверенностью сказать, что сегодня язык разметки HTML является основой всех размещенных в Интернете электронных документов. Он выступает в роли некоего фундамента, на базе которого реализуются прочие сетевые программные технологии, призванные в конечном итоге повысить общую привлекательность, эффективность и интерактивность носителей информационных данных в Сети. Так же как обычный фундамент строящегося дома представляет собой несущую конструкцию, на которой впоследствии возводятся стены, HTML служит простейшим и по большому счету незаменимым средством создания гипертекстовых документов, которое в совокупности с более гибкими и функциональными интернет-технологиями позволяет получать воистину фантастические результаты.

HTML постоянно набирает популярность, причем не только в сфере интернет-технологий, но и в области предоставления презентационных услуг, рекламно-выставочной деятельности, внедряется в состав программного обеспечения и пр. Корпоративные клиенты все чаще разрабатывают CD-презентации и демонстрационные ролики, заказывают электронные визитки и рекламные обращения. Частный пользователь все больше склоняется к мысли о составлении интерактивных портфолио и резюме, позволяющих, в отличие от листа бумаги, ярче и привлекательнее преподнести свои знания и умения потенциальному работодателю. Разработчики ПО (программного обеспечения) делают информационные и рекламные вставки справочного характера внутри создаваемых ими программных пакетов. И все это так или иначе реализуется с помощью простого и доступного, но вместе с тем эффективного языка разметки HTML.

Спецификация HTML

Многие называют HTML языком программирования. Это не совсем верно, так как в традиционном понимании HTML является языком *разметки* электронных документов, лишь указывающим программам просмотра HTML-страниц (браузерам) форму представления описанной в документе информации.

Начиная с середины 90-х годов XX века HTML претерпел некоторые изменения в своей спецификации — варианты используемых инструкций, применяемых тегов и модулей горячо обсуждались и изменялись. На сегодня последней версией языка HTML является версия 4.01.

Начиная с момента своего возникновения разработкой спецификации языка HTML стала заниматься организация под названием "Консорциум W3C" (World Wide Web Consortium). Ее основной задачей являлось составление и принятие технических рекомендаций единого стандарта разметки гипертекстовых документов. Практическая необходимость работы над стандартом была обусловлена постоянным ростом популярности Интернета, в рамках которого производители программ (браузеров) для просмотра Web-документов выдвигали свои предложения по улучшению правил описания гипертекстовых данных.

Версия HTML 4.0, выпущенная Консорциумом в середине 1997 года и ставшая признанной спецификацией в конце того же года, является на сегодня последней номерной версией языка разметки HTML. Некоторые несущественные дополнения, внесенные в спецификацию в декабре 1999 года, мало повлияли на структуру самого языка, а версия получила небольшое добавление в виде цифры 1, т. е. стала называться 4.01.

Хочется добавить, что деятельность Консорциума по сути призвана регулировать и контролировать развитие и совершенствование языка гипертекстовой разметки HTML, учитывая потребности сферы интернет-технологий и компаний-разработчиков, работающих на рынке браузеров. Однако в действительности ситуация не такая однозначная — производители программ для просмотра HTML-документов постоянно предлагают технологические нововведения в спецификацию языка, часть которых получает одобрение Консорциума. Остальная часть инноваций, не вошедшая в состав принятой W3C рекомендации, может тем не менее внедряться в программную платформу выпускаемых браузеров, что на практике вызывает проблемы несовместимости электронных документов при их просмотре браузерами разных моделей и версий.

Какой браузер выбрать

Специальные программы для просмотра электронных документов, созданных по правилам языка разметки HTML, называются браузерами. Основная функция браузера заключается в интерпретации кода HTML и выводе визу-

ального результата на экран монитора пользователя. Сегодня существует большое количество самых разнообразных браузеров, однако наибольшей популярностью пользуются лишь три программы: Internet Explorer, Netscape Navigator и Opera. Рассмотрим немного подробнее каждый из этих браузеров.

Internet Explorer

В августе 1995 года компания Microsoft выпустила очередное обновление своей операционной системы Windows 95, в состав которой впервые был включен интернет-браузер Internet Explorer 1.0. Однако первоначальный программный код браузера принадлежал не Microsoft, а компании Spyglass, которая впоследствии продала лицензию на изменение и коммерческое распространение кода разработчикам Microsoft. Последние поставили Internet Explorer на более высокий уровень развития, что позволило браузеру-дебютанту составить достойную конкуренцию Netscape Navigator — браузеру, которым к 1995 году пользовались свыше 50% посетителей Интернета.

В конце того же года Microsoft выпускает окончательный и доработанный вариант Internet Explorer 2.0, а весной 1996 года появляется версия 3.0, содержавшая для того времени целый ряд нововведений, таких как поддержка фреймов, подключение программных надстроек (plug-in) и пр.

Выход четвертой версии Internet Explorer (вторая половина 1997 года) положил начало стремительному увеличению доли рынка браузеров корпорации Microsoft (к концу года по самым разным данным она охватывала 60% рынка, в то время как ее основной соперник, компания Netscape, — всего около 30%).

Однако Microsoft не останавливается на достигнутом превосходстве. В 1999 году выходит пятая версия популярного во всем мире браузера, в котором были реализованы новые и усовершенствованы старые технологии. Примерно через год появляется версия 5.5 и, наконец, в октябре 2001 года свет увидела последняя (на момент написания книги) версия — 6.0.

Можно назвать несколько основных возможностей браузера, благодаря которым Internet Explorer получил такую популярность:

- быстрый запуск программы;
- поддержка многих технологий, не реализованных или реализованных недостаточно в других браузерах (ActiveX, CSS1/CSS2, "плавающие фреймы" и др.);
- демократичность интерпретации HTML-кода. При загрузке документа, код которого содержит незнакомые конструкции и/или ошибки, Internet Explorer в большинстве случаев просто не выводит часть, вызывающую затруднения, на экран, в то время как, например, браузер Netscape может отобразить структуру такого документа нарушенной или вообще ничего не вывести на экран монитора;

- полная интеграция с другими приложениями Microsoft, работающими под управлением ОС Windows;
- возможность масштабирования графических изображений, открытых в отдельном окне.

Среди недостатков можно выделить следующие:

- нестабильность работы;
- среднюю скорость загрузки информации.

Netscape

Компания Netscape Communications Corporation практически с самого начала считалась основным конкурентом корпорации Microsoft в продвижении своего браузера.

Современный браузер Netscape берет свое начало в марте 1993 года, когда один из основателей будущей компании Марк Андриссен (Marc Andreessen) — анонсирует выход программы Mosaic (прототипа будущего браузера Netscape). На следующий год Марк Андриссен и его коллега Джим Кларк (Jim Clark) основывают компанию Mosaic Communications (будущая Netscape Communications Corporation), а буквально через несколько месяцев на свет появляется первая версия интернет-браузера Netscape 0.9.

Компания расширяется, Netscape получает распространение, и в итоге к лету 1995 года большинство посетителей Всемирной паутины (около 80%) используют для путешествия по Интернету именно эту программу. Однако вскоре появляется Internet Explorer, который начинает всерьез конкурировать с Netscape, и основатели компании выпускают новую версию своего браузера (2.0), снабдив его не только новым именем Netscape Navigator, но и некоторыми техническими возможностями, тем самым начав знаменитую "войну браузеров".

В конце 1998 года крупнейший интернет-провайдер Соединенных Штатов Америки America-On-Line (AOL) покупает компанию Netscape и все права на развитие одноименного браузера. В 2000 году выходит Netscape 6.0, параллельно с выходом которого анонсируется проект Mozilla 0.6. Оба приложения на тот момент использовали единое программное ядро Gecko, но Netscape как торговая марка принадлежал AOL, а Mozilla разрабатывался в качестве независимого проекта.

Наконец, в августе 2002 года выходит последняя на момент написания книги версия Netscape — 7.0, а следом за ней появляется Mozilla 1.0.

Основными преимуществами браузеров семейства Netscape являются:

- сравнительно небольшой размер программы;
- предоставление пользователю расширенного управления содержанием электронных документов;

- улучшенная организация внутренних данных;
- поддержка технологии смены skin'ов (изменение внешнего вида программы в соответствии с выбранной оформительской схемой).

Недостатков тоже хватает, учитывая трудный путь развития, который прошли браузеры Netscape:

- отсутствие поддержки некоторых интерактивных технологий, рекомендованных Консорциумом W3C;
- низкая скорость работы;
- слишком долгое время запуска программы.

Opera

Компания Opera Software (г. Осло, Норвегия) разработала одноименный браузер в 1994 году для норвежской телекоммуникационной компании Telenor. Группа разработчиков, включавшая в себя двух основателей Opera Software, Йона Штефенсона фон Тежнера (Jon Stephenson fon Tetzchner) и Гера Иварсоя (Geir Ivarsoy), поставила перед собой задачу создать интернет- и мультимедиа-приложение, которым могли бы пользоваться все желающие, независимо от системных возможностей своих компьютеров.

В первоначальную концепцию браузера были заложены такие критерии, как скорость запуска программы и загрузки информации, небольшой размер приложения, минимальные требования к ресурсам компьютера пользователя.

Программа, изначально задуманная как небольшой по размеру быстрый браузер для компьютеров с незначительными ресурсами, какое-то время использовалась в пределах внутренней информационной сети компании Telenor, а к концу 1995 года авторы Opera покинули стены компании, чтобы продолжить самостоятельное развитие своего детища. Наконец, во второй половине 1996 года браузер Opera 2.1 стал доступен для загрузки в Интернете в качестве 90-дневной условно-бесплатной (Shareware) версии.

К числу основных отличий Opera от других браузеров, которые с полной уверенностью можно считать преимуществами, отнесем следующие:

- небольшой размер;
- минимальные системные требования;
- быструю скорость загрузки HTML-документов;
- расширенные настройки;
- высокую масштабируемость просматриваемого документа.

Однако и у Opera есть ряд недостатков, которые также следует упомянуть:

- статус коммерческого программного продукта (регистрация стоит 39 долларов для полной версии; Freeware-вариант будет постоянно "радовать" вас показом чужих рекламных баннеров);

- отсутствие поддержки некоторых русских кодировок в английских версиях программы;
- недостаточно высокий уровень надежности выполнения скриптов на стороне пользователя (JavaScript/VBScript).

На момент написания книги последней версией браузера Opera стала версия 7.0.

Как видите, каждый из описанных интернет-браузеров прошел долгую историю развития, имеет свои плюсы и минусы, получает одобрение и порицание, обладает сторонниками и противниками своих функциональных возможностей. В конечном итоге, выбор браузера, с которым вы будете работать в процессе освоения языка гипертекстовой разметки HTML, остается за вами.

В заключение лишь приведу статистику использования браузеров. Согласно подсчетам известной исследовательской группы OneStat, в 2002 году около 94,6% пользователей во всем мире выходило в Интернет с помощью Internet Explorer, в то время как на долю Netscape пришлось всего 3,3%. Остальные браузеры отстают безнадежно пользователей Mozilla 1.0 и Opera 6.0 насчитывается всего около 0,8%.

Статистика использования браузеров в российской части Интернета почти повторяет мировые показатели: Internet Explorer — 91,5%, Netscape — 3,2%, а прочие браузеры — 5,2% (источник: ArtLebedev Group, ноябрь 2002 года).

Структура HTML-документа

Как уже было сказано ранее, язык HTML представляет собой набор специальных правил. Каждому правилу соответствует свое название, свойство и значение. Например, чтобы задать правило жирного начертания обыкновенного текста, необходимо использовать следующую HTML-конструкцию:

```
<B>Обыкновенный текст</B>
```

Как видно из примера, текст, который должен отображаться жирным начертанием, обособлен группами символов <в> и </в>. Такие группы принято называть тегами. Сразу следует сказать, что теги бывают одинарными и парными. В случае с нашим примером тег </в> является парным, т. к. он закрывает HTML-конструкцию вместе с символом "/" (прямой слэш). Иногда теги, которые необходимо закрывать парным тегом, называют тегами-контейнерами.

Структура любого тега подразумевает указание самого тега, его параметра и значения этого параметра. При этом наименование параметра и его значения может писаться как строчными, так и прописными буквами. Значение параметра ставится в кавычки.

Кроме того, параметры и их значения могут либо вообще отсутствовать у какого-то определенного тега, либо считаться необязательными и устанавливаемыми только в случае изменения стандартных характеристик тега:

```
<TABLE WIDTH="100%" BORDER>
```

В приведенном примере для тега построения таблицы <TABLE> один параметр (WIDTH) указан со значением, другой (BORDER) такового не содержит.

Некоторые теги могут указываться вообще безо всяких параметров (например, принудительный перенос строки
). Другие теги могут включать несколько значений одного параметра ("INDEX, FOLLOW" для метаопределения "robots").

Любой HTML-документ содержит три основных обязательных раздела: HTML, HEAD и BODY. Рассмотрим подробнее каждый из них.

Раздел *HTML*

Раздел HTML определяет специфику документа, содержание которого будет интерпретироваться браузером. Раздел описывается тегом-контейнером <HTML></HTML> и дает браузеру информацию о том, что документ разработан с помощью языка разметки HTML. В предыдущих версиях языка тег <HTML> использовался с некоторыми параметрами, однако спецификация 4.0 отменила это правило (в принципе, сегодня большинство браузеров способно распознать HTML-документ и без указания данного тега, тем не менее, пропускать раздел HTML разработчикам не рекомендуется).

Раздел *HEAD*

Раздел HEAD выполняет функцию рабочего заголовка HTML-документа и является, по сути, "бойцом невидимого фронта" - - теги, указываемые внутри этого раздела, чрезвычайно важны и могут сильно влиять на внешний вид документа, но сами остаются незаметными глазу пользователя. Данному разделу сопоставлен парный тег <HEADX/HEAD>.

Рассмотрим теги HTML, которые указываются внутри раздела HEAD.

Название документа *<TITLE>*

Парный тег <TITLEX/TITLE> предназначен для указания имени созданному электронному документу. Следует помнить, что под именем документа в данном случае имеется в виду не файловое наименование, а визуальный заголовок HTML-страницы.

Указание конструкции <TITLEX/TITLE> не является обязательным, однако рекомендуется по ряду причин:

- отсутствие тега названия документа заставит браузер при интерпретации HTML-кода вывести в заголовке окна фразу типа **Untitled Document**

(Документ без названия), что не соответствует ни тематике вашего электронного документа, ни его наполнению;

- при попытке добавить созданный вами HTML-документ (без тега-контейнера <TITLEX/TITLE>) в "закладки" браузера пользователю придется самостоятельно вписывать название добавляемой страницы;
- поисковые системы, столкнувшись с безымянной страницей, занесут ее в свои базы данных под заголовком Untitled, что сделает HTML-документ безликим и похожим на миллионы других электронных документов, размещенных в Интернете.

Связь между документами <*LINK*>

Часто бывает так, что несколько различных документов структурно или функционально взаимосвязаны между собой. В этом случае используется тег <LINK>, который не является парным, а значит не требует указания закрывающего тега:

```
<LINK HREF="styles/main.css" TYPE="text/css" REL="stylesheet">
```

Приведенный пример показывает отношение между HTML-документом, в разделе HEAD которого указана данная кодовая конструкция, и внешним файлом шаблона стилей `main.css`.

Возможные параметры тега <LINK> приведены в табл. 1.1.

Таблица 1.1. *Возможные параметры тега <LINK>*

Параметр	Назначение
HREF	Указание ссылки на внешний документ
TYPE	Тип внешнего документа
REL	Отношение между текущим и внешним документом
REV	Отношение между внешним и текущим документом

Мета-определения <*META*>

Мета-определения электронного документа описываются целым рядом параметров, входящих в состав непарного тега <META> и предназначеных для описания внутренних свойств HTML-файла.

Все мета-определения, в сущности, имеют два основных типа данных: HTTP-EQUIV и NAME. Первый можно отождествить с заголовками протокола передачи гипертекстовых данных HTTP — (HyperText Transfer Protocol). Структура мета-тега этого типа выглядит так:

```
<META HTTP-EQUIV="имя" CONTENT="содержание">
```

Рассмотрим кодовые конструкции по данному типу мета-определений (с использованием соответствующих параметров HTML).

META HTTP-EQUIV="expires"

Конструкция определяет возможность запрета кэширования HTML-страниц за счет указания даты устаревания документа (актуальна для интернет-ресурсов с динамически изменяющимся содержанием). По истечении срока, указанного в этом параметре, браузер при повторном обращении к документу будет перенаправлен к источнику для обновления информации. Значение параметра CONTENT описывается в следующей последовательности: день недели (Mon, Tue, Wed и т. д.), число (01, 02, 03 и т. д.), месяц (Jan, Feb, Mar и т. д.), год, время (часы, минуты, секунды) и часовой пояс (GMT):

```
<META HTTP-EQUIV="expires" CONTENT="Sat, 25 Jan 2003 15:30:00 GMT">
```

Приведенный пример говорит о том, что в субботу, 25 января 2003 года, в 15 часов 30 минут по Гринвичу "срок годности" HTML-документа истечет, что потребует от браузеров обновления данных при обращении к содержанию страницы.

META HTTP-EQUIV="refresh"

Такая конструкция указывает браузеру перезагружать содержимое окна через заданный промежуток времени. При добавлении дополнительного параметра URL через указанное время (в секундах) будет произведена переадресация на внешний адрес. Следующий пример дает браузеру команду ровно через 10 секунд перенаправить посетителя по адресу **http://www.site.ru:**

```
<META HTTP-EQUIV="refresh" CONTENT="10; URL=http://www.site.ru">
```

Причин для использования такой конструкции может быть несколько: ваш Web-сайт переехал на другой сервер, вы хотите показать рекламный блок, а потом перенаправить посетителя на заглавную страницу и т. д. Простая перезагрузка содержимого страницы может пригодиться в случае динамического изменения информации, работы скрипта по случайному выводу данных, электронных презентаций и пр.

META HTTP-EQUIV="Content-Type"

Главная функция этой конструкции — определение типа и кодировки документа. Наиболее используемыми значениями кодировки русскоязычных HTML-документов являются Windows-1251 и KOI8-R. На некоторых серверах возможно автоматическое перекодирование документов, в связи с чем использовать эту конструкцию не рекомендуется. Также необходимо проследить, чтобы кодировка текста документа совпадала с кодировкой, обозначенной в значении charset.

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Windows-1251">
```

Упомянутые и прочие малоиспользуемые конструкции типа META HTTP-EQUIV перечислены в табл. 1.2.

Таблица 1.2. *Параметры типа мета-определений HTTP-EQUIV*

Параметр	Функция	Подпараметры
expires	Запрет кэширования HTML-документа по истечении указанного срока	—
refresh	Перезагрузка/переадресация через заданный промежуток времени	URL
Content-Type	Определение типа и кодировки документа	charset
Content-Language	Указание языка документа	—
Cache-Control	Контроль кэширования документа	—
Window-target	Определение места загрузки документа (используется для фреймовых структур)	—
Pragma	Управление кэшированием документа по протоколу HTTP/1.0	—
Set-Cookie	Настройка чтения данных Cookies (данных пользовательского компьютера)	name expires domain path secure
Ext-cache	Управление альтернативным кэшем документа	name instructions
Location	Указание места расположения документа в Интернете (полный адрес)	—

Теперь рассмотрим основные конструкции мета-определений второго типа — NAME.

META NAME="Keywords"

Конструкция задает набор ключевых слов документа, предназначенных для индексирования поисковыми системами (добавления информации о документе в базы данных). Слова указываются через запятую:

```
<META NAME="keywords" CONTENT="HTML, Web, WWW, Web-страница, Интернет">
```

META NAME="Description"

Предоставление небольшого описания текущего HTML-документа, также необходимого для поисковых систем (при поиске информации описание выводится рядом со ссылкой на найденный интернет-ресурс):

```
<META NAME="description" CONTENT="HTML. Экспресс-курс: Быстрое освоение популярного языка гипертекстовой разметки HTML">
```

META NAME="Author"

Конструкция предназначена для указания автора (авторов) текущего документа.

```
<META NAME="Author" CONTENT="Алексей Петюшкин">
```

META NAME="Robots"

Это одна из наиболее важных конструкций, используемая при описании правил для индексирования документа поисковыми системами (роботами). Отдельно следует перечислить возможные значения параметра CONTENT:

- ALL — разрешение индексирования документа со всеми присутствующими в нем гиперсвязями (ссылками);
- NONE — запрет индексирования документа со всеми присутствующими в нем гиперсвязями;
- INDEX — разрешение индексирования документа;
- NOINDEX — запрет индексирования документа;
- FOLLOW — разрешение индексирования присутствующих в документе гиперсвязей;
- NOFOLLOW — запрет индексирования присутствующих в документе гиперсвязей.

Если конструкция META NAME="Robots" не указывается, то поисковый робот индексирует документ со всеми присутствующими в нем гиперссылками (равнозначно значению ALL или INDEX, FOLLOW).

```
<META NAME="robots" CONTENT="ALL">
```

Эти и некоторые другие конструкции типа NAME перечислены в табл. 1.3.

Таблица 1.3. Параметры типа мета-определений NAME

Параметр	Функция	Подпараметры
keywords	Указание ключевых слов документа	—
description	Указание краткого описания документа	—
author	Информация об авторе (авторах) документа	—

Таблица 1.3 (окончание)

Параметр	Функция	Подпараметры
robots	Составление правил для индексирования документа поисковыми системами (роботами)	ALL NONE INDEX NOINDEX FOLLOW NOFOLLOW
rating	Установление возрастной категории содержания документа	—
document-state	Определение частоты индексирования документа	—
revisit	Определение промежутка времени, через которое должно производиться повторное индексирование документа	—
distribution	Категория распространения документа (массовое или ограниченное)	—
URL	Указание основного адреса документа для индексирования (в случае существования "зеркальных" копий)	—
copyright	Установление авторского права и перечисление условий распространения документа	—
generator	Информация о программном обеспечении, с помощью которого создавался документ	—

Следует помнить о том, что использование абсолютно всех мета-определений не нужно. В зависимости от типа документа, его содержания и прочих факторов, надо выбрать только самые необходимые теги мета-данных. Наиболее важными являются данные для поисковых систем и указание кодировки документа.

Элементы <*STYLE*> и <*SCRIPT*>

В структуру раздела документа HEAD также могут входить теги-контейнеры <STYLEX/STYLE> И <SCRIPTX/SCRIPT>. Первый парный тег описывает стилевые шаблоны документа, второй — содержит код исполняемых сценариев (скриптов).

Раздел *BODY*

Раздел BODY является одним из самых важных компонентов любого HTML-документа, т. к. в нем располагается содержательная часть, которая выводится браузером на экран монитора пользователя.

Раздел описывается парным тегом <BODYX/BODY>, внутри которого размещается большинство существующих тегов HTML. Тег <BODY> имеет ряд параметров (табл. 1.4), которые условно можно разделить на четыре основные группы (параметры фона, границ документа, текста и гиперссылок).

***Таблица 1.4.** Параметры тега <BODY>*

Параметр	**Функция**
BGCOLOR	Определение цвета фона
BACKGROUND	Указание фонового рисунка
BGPROPERTIES	Изменение свойств фона (например, фиксирование фонового рисунка)
TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN, MARGINWIDTH, MARGINHEIGHT	Определение размера отступов
TEXT	Определение цвета основного текста
LINK, ALINK, VLINK	Определение цвета гиперссылок

Параметры фона

Параметрами фона документа являются BGCOLOR, BACKGROUND и BGPROPERTIES. BGCOLOR устанавливает цвет фона, значение которого может быть введено в символьном эквиваленте, в шестнадцатеричном коде или в формате цветовой модели RGB. Система указания цвета в HTML основана на трех основных цветах: красном, зеленом и синем (модель RGB — Red, Green, Blue). Любое значение RGB может быть преобразовано в шестнадцатеричный формат (от 00 до FF с приставкой # (читается "диез")). Некоторым значениям упомянутых моделей соответствует символьное название цвета. Таким образом, один и тот же цвет можно указать тремя возможными способами.

Далее показано три варианта установки цвета фона (белого):

- `<BODY BGCOLOR="white">`
- `<BODY BGCOLOR="#FFFFFF">`
- `<BODY BGCOLOR="255,255,255">`

Параметр BACKGROUND позволяет накладывать на фон документа графическое изображение:

```
<BODY BACKGROUND="images/bg.gif">
```

Параметр BGPROPERTIES поддерживается только браузером Microsoft Internet Explorer и позволяет менять свойства фона документа. Например, конструкция

```
<BODY BACKGROUND="images/bg.gif" BGPROPERTIES="fixed">
```

позволит прокручивать содержание документа, оставляя фоновое графическое изображение в зафиксированном виде.

Описанные параметры не являются обязательными, однако использование *BGCOLOR* рекомендуется по следующей причине: пользователь в настройках своего браузера может поставить любой цвет фона, а разработчик, полагая, что белый цвет является основным по умолчанию, может не указать этот параметр. В результате вместо подразумевающегося белого цвета, фон может оказаться черным, зеленым и т. д., что способно привести к нарушению оформления документа. Также наряду с графическим изображением фона рекомендуется использовать и параметры цвета на тот случай, если рисунок не загрузится (тогда браузер отобразит цвет).

Параметры границ документа

Параметры границ HTML-документа создают отступы заданного размера от верхнего, нижнего, левого и правого краев документа. Этим отступам соответствуют параметры TOPMARGIN, BOTTOMMARGIN, LEFTMARGIN и RIGHTMARGIN. Значения для них задаются в пикселах:

```
<BODY TOPMARGIN="5" BOTTOMMARGIN="5" LEFTMARGIN="10" RIGHTMARGIN="10">
```

Однако данные параметры не воспринимаются некоторыми браузерами. В частности, Netscape объединяет упомянутые параметры в две группы — горизонтальные и вертикальные отступы:

```
<BODY MARGINWIDTH="10" MARGINHEIGHT="5">
```

Таким образом, если вы хотите учесть особенности всех браузеров, то в HTML-конструкцию надо подставить и те, и другие параметры:

```
<BODY TOPMARGIN="5" BOTTOMMARGIN="5" LEFTMARGIN="10" RIGHTMARGIN="10"
MARGINWIDTH="10" MARGINHEIGHT="5">
```

Параметры текста

Из параметров текста документа реально применяется только один — TEXT. Он задает цвет основного текста на странице (значение параметра может быть введено аналогично цвету фона документа):

```
<BODY TEXT="black">
```

Параметры гиперссылок

Параметры гиперссылок (связей с внутренними или внешними документами) определяют цвет активных (ALINK), не посещенных (LINK) и посещенных (VLINK) ссылок:

```
<BODY LINK="#0000FF" ALINK="#0000FF" VLINK="blue">
```

Первый HTML-документ

Итак, мы рассмотрели основные разделы любого HTML-документа. В целом, код простейшей страницы представлен в листинге 1.1.

Листинг 1.1. Пример простейшего HTML-документа

```
<HTML>
  <HEAD>
    <TITLE>Мой первый HTML-документ</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=Windows-1251">
<META NAME="keywords" CONTENT="HTML, документ, первый">
<META NAME="description" CONTENT="Это мой первый HTML-документ">
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TOPMARGIN="30" TEXT="black" LINK="#00FF00"
ALINK="#00FF00" VLINK="blue">
Это мой первый HTML-документ!
</BODY>
</HTML>
```

Теперь сохраните код под любым именем, не забыв присвоить файлу расширение html или htm. Откройте файл в браузере. Это ваш первый HTML-документ!

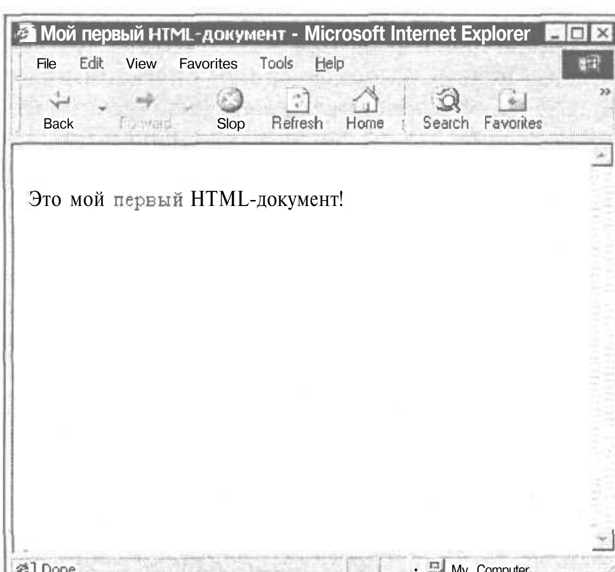


Рис. 1.1. Ваш первый HTML-документ

Глава 2

Работаем с текстом

Форматирование текста

Можно с полной уверенностью заявить, что текст занимает далеко не последнее место в структуре HTML-документа. В большинстве случаев мы посещаем интернет-сайты в поисках именно текстовой информации: статей и художественных произведений, новостей и технической документации и т. п. Поэтому от того, как оформлен текст, как он преподнесен посетителю, зависит многое в судьбе любого Web-сайта.

Область работы с текстом в HTML включает в себя большое количество всевозможных тегов, составляющих две основные группы — теги логического форматирования и теги физического форматирования.

В группу тегов логического форматирования входят теги, отображающие на экране монитора элементы документа таким образом, как установлено по умолчанию в спецификации языка разметки HTML. Переопределить их параметры или свойства нельзя, за исключением ситуаций использования стилевых шаблонов CSS и обособления тегами физического форматирования. Результат действия разных тегов логического форматирования визуально может совпадать, ибо основное их предназначение заключается в логическом выделении отдельных элементов HTML.

Теги физического форматирования позволяют разработчику HTML-документа визуально изменять вид текста, варьируя его параметры и значения. Другими словами, теги физического форматирования предназначены для выделения отдельных текстовых фрагментов различными способами, установленными автором документа.

Рассмотрим подробнее наиболее используемые теги из каждой группы.

Теги логического форматирования

Тег `<ACRONYM>` — используется для расшифровки аббревиатур. Реализуется через параметр TITLE (листинг 2.1) и отображается в браузере при наведении курсора на слово-аббревиатуру (рис. 2.1).

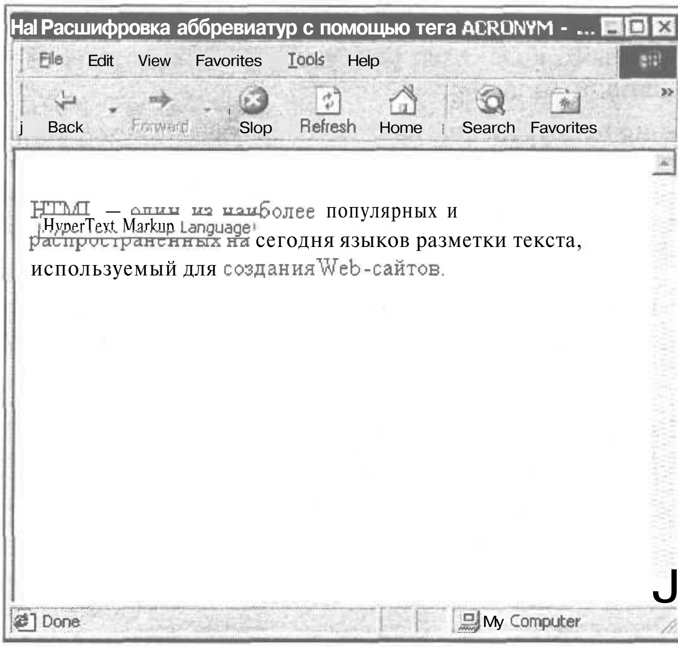


Рис. 2.1. Использование тега <ACRONYM>

Листинг 2.1. Использование тега <ACRONYM>

```
<HTML>
  <HEAD>
    <TITLE>Расшифровка аббревиатур с помощью тега ACRONYM</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<ACRONYM TITLE="HyperText Markup Language">HTML</ACRONYM> - один из наи-
более популярных и распространенных на сегодня языков разметки текста,
используемый для создания Web-сайтов.
</BODY>
</HTML>
```

Помимо <ACRONYM>, возможность отображения слова в сокращенном виде и подсказки к нему существует у тега <ABBR>.

Тег <CITE> — призван выделять различные цитаты и высказывания, названия библиографических источников и пр. Текст, помещенный между тегами <CITE> и </CITE>, браузером выделяется курсивом. Визуально аналогичен тегам <EM> и <I>.

Тег <CODE> — предназначен для визуального выделения небольших фрагментов программного кода. Код, размещенный в этом парном теге, отображается моноширинным шрифтом.

Тег <DEL> — используется для обозначения удаленного текста. Имеет необязательные для указания параметры DATETIME и CITE (первый показывает дату удаления, второй ссылается на источник причины удаления). Текст, заключенный между тегами <DEL>, отображается зачеркнутым и аналогичен действию тегов <s> или <STRIKE>.

Тег <EM> -- используется для интонационного выделения определенного текстового фрагмента (листинг 2.2). Браузерами отображается курсивом (так же, как теги <CITE> и <I>) (рис. 2.2).

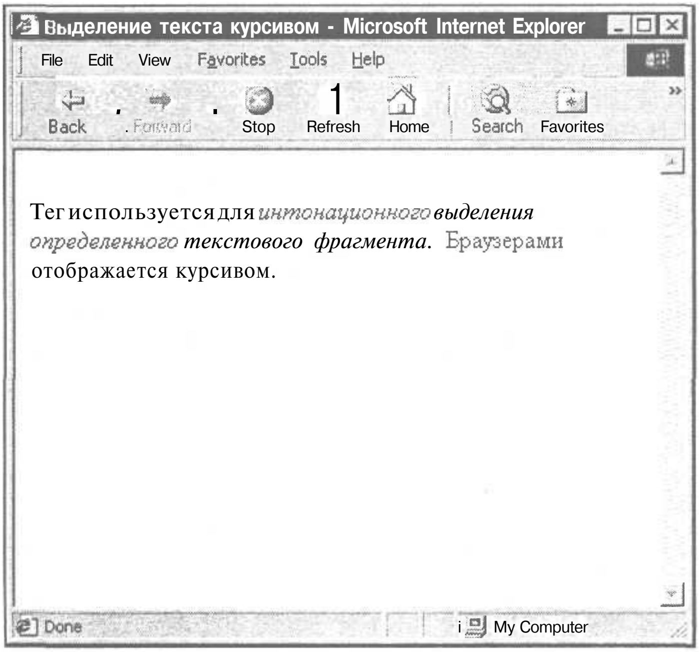


Рис. 2.2. Использование тега <EM>

Листинг 2.2. Использование тега <EM>

```
<HTML>
  <HEAD>
    <TITLE>Выделение текста курсивом</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
```

```
VLINK="blue">
Тег используется для <EM>интонационного выделения определенного текстово-
го фрагмента</EM>. Браузерами отображается курсивом.
</BODY>
</HTML>
```

Теги <H1>, <H2>... <нб> — тег <н> с указанием одной из цифр (от 1 до 6) задает определенный размер заголовка всего текста целиком или его конкретного фрагмента (листинг 2.3). Соответственно, заголовок <H6> будет минимальным, а <H1> — самым большим (рис. 2.3). Особенностью тегов <н> является то, что они уже подразумевают отступ от текстовой части и разработчикам HTML-документов не приходится делать отступ самостоятельно.

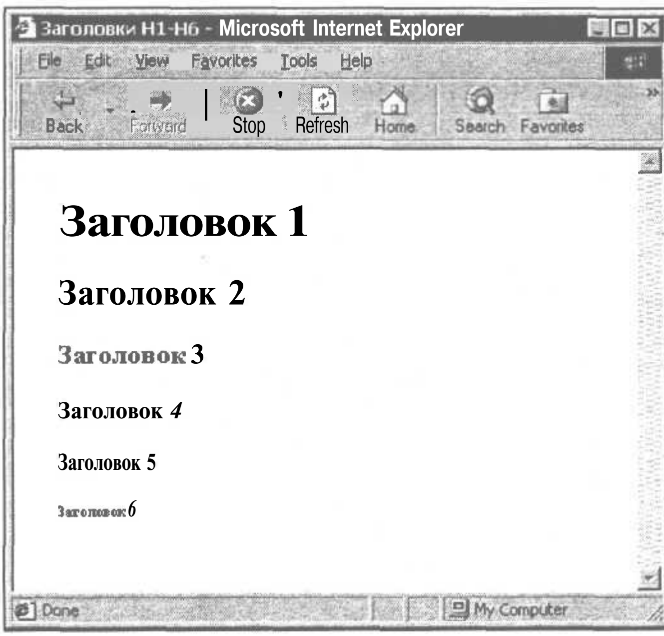


Рис. 2.3. Использование тегов <H1>—<H6>

Листинг 2.3. Использование тегов <H1>—<H6>

```
<HTML>
  <HEAD>
    <TITLE>Заголовки H1-H6</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
```

```
<H1>Заголовок 1</H1>
<H2>Заголовок 2</H2>
<H3>Заголовок 3</H3>
<H4>Заголовок 4</H4>
<H5>Заголовок 5</H5>
<H6>Заголовок 6</H6>
</BODY>
</HTML>
```

Тег **<STRONG>** — используется для выделения фрагментов текста, на которых необходимо акцентировать внимание пользователя (листинг 2.4). Браузерами результат такого форматирования отображается жирным начертанием (рис. 2.4). В визуальном плане аналогичен тегу <в>.

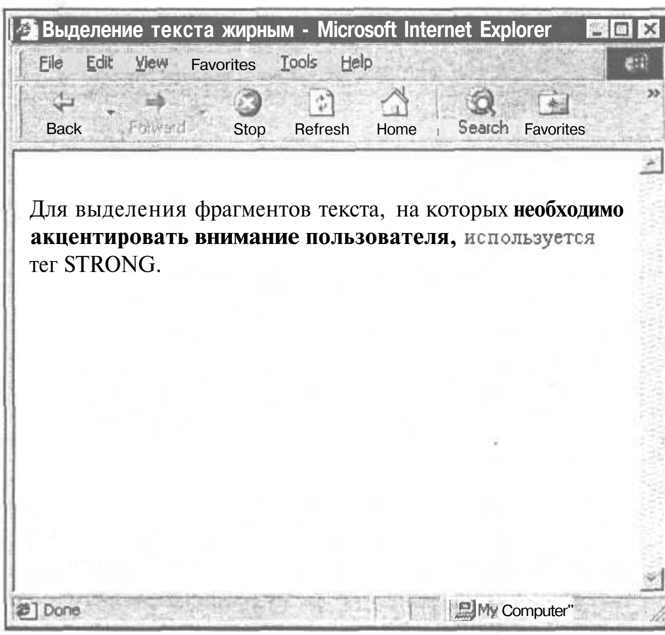


Рис. 2.4. Использование тега <STRONG>

Листинг 2.4. Использование тега <STRONG>

```
<HTML>
  <HEAD>
    <TITLE>Выделение текста жирным</TITLE>
  </HEAD>
```

```
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
Для выделения фрагментов текста, на которых <STRONG>необходимо
акцентировать внимание пользователя</STRONG>, используется тег STRONG.
</BODY>
</HTML>
```

Полный перечень рассмотренных тегов логического форматирования приведен в табл. 2.1.

***Таблица 2.1.** Теги логического форматирования*

Тег	Функция
ACRONYM	Расшифровка аббревиатур
CITE	Указание цитаты
CODE	Указание фрагмента программного кода
DEL	Обозначение удаленной информации
EM	Выделение курсивом
H1-H6	Создание заголовков
STRONG	Выделение жирным начертанием

Тегифизическогоформатирования

Тег <в> — функция этого тега аналогична действию тега <STRONG>, а именно — визуальное выделение фрагмента текста, на котором следует сделать акцент, отображается жирным начертанием. В последней версии спецификации HTML вместо <в> рекомендуется использование тега логического форматирования <STRONG>.

Тег **<BASEFONT>** — предназначен для определения типа шрифта, а также его цвета и размера, которые будут считаться принятыми по умолчанию для всего HTML-документа. Параметры тега <BASEFONT> аналогичны параметрам тега <FONT> и могут быть изменены по ходу текста дополнительным форматированием с помощью тега <FONT>.

Следует заметить, что тег <BASEFONT> используется в двух разделах документа: HEAD и BODY. Закрывающего тега <BASEFONT> не требует.

Тег <BIS> -- используется, если необходимо выделить часть текста небольшим увеличением размера шрифта относительно остальных слов. Однако следует заметить, что спецификация HTML не одобряет подобного подхода и рекомендует применять теги заголовков <н>.

Тег **<BLINK>** — поддерживается только браузером Netscape и реализует мигающий эффект выделенного фрагмента текста:

Тег <FONT> — один из основных тегов физического форматирования текста, отображающий свойства шрифтов. Для него могут использоваться следующие параметры:

- FACE -- параметр, посредством которого браузер отображает текст указанным в HTML-коде шрифтом. При этом значение параметра FACE должно соответствовать установленному на компьютере пользователя шрифту. В случае если такого шрифта нет, текст будет показан стандартным (по умолчанию) шрифтом.

  Часто разработчики HTML-документов добавляют сразу несколько значений параметра FACE. Таким образом, если первого шрифта не окажется на компьютере пользователя, то браузер отобразит текст вторым шрифтом. Не будет найдено второго шрифта, браузер приступит к поиску третьего и т. д. (рис. 2.5, листинг 2.5).

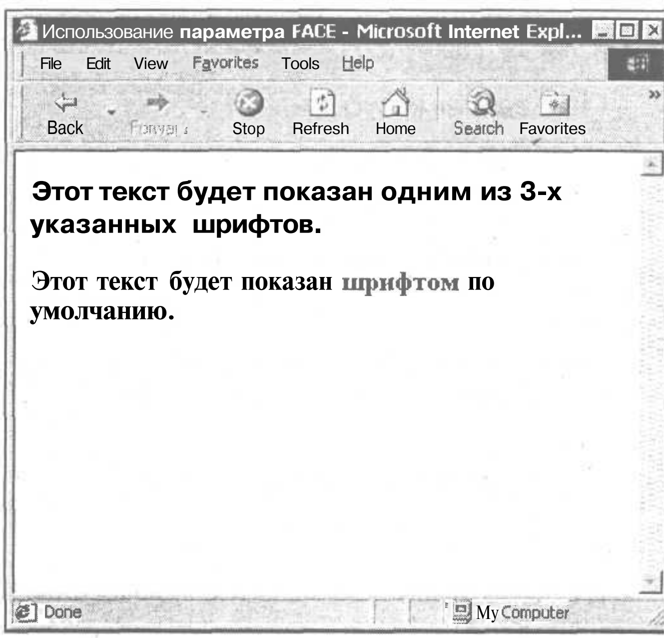


Рис. 2.5. Использование параметра FACE тега <FONT>

Листинг 2.5. Использование параметра FACE тега <FONT>

```
<HTML>
  <HEAD>
    <TITLE>Использование параметра FACE</TITLE>
  </HEAD>
```

```
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">

<H3>

<FONT FACE="Tahoma", "Arial", "Verdana">Этот текст будет показан одним из
3-х указанных шрифтов.</FONT>

</H3>

<H3>

<FONT FACE="Allegro">Этот текст будет показан шрифтом по умолча-
нию.</FONT>

</H3>

</BODY>
</HTML>
```

- COLOR — параметр, предназначенный для выбора цвета, которым будет написан текст. HTML-конструкция имеет следующий вид:

  ```
  <FONT COLOR="red">Этот текст будет написан красным цветом.</FONT>
  ```

- SIZE — функцией этого параметра является определение размера шрифта, которое происходит по условной шкале от 1 до 7. При этом указание размера может осуществляться как в виде относительного показателя (+1), так и в виде абсолютного (1). Средним значением размера шрифта принято считать показатель 3. Однако окончательный вид шрифта с учетом свойств его размеров может отличаться в зависимости от модели и версии браузера.

Наконец, возможно использование сразу всех параметров тега <FONT> в пределах одной HTML-конструкции (листинг 2.6, рис. 2.6).

Листинг 2.6. Использование всех параметров тега <FONT>

```
<HTML>

  <HEAD>

    <TITLE>Использование всех параметров тега FONT</TITLE>

  </HEAD>

<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">

<FONT FACE="Tahoma", "Helvetica" COLOR="#000000" SIZE="2">Этот текст бу-
дет показан одним из 3-х указанных шрифтов, черным цветом и размером
"1".</FONT>

</BODY>

</HTML>
```

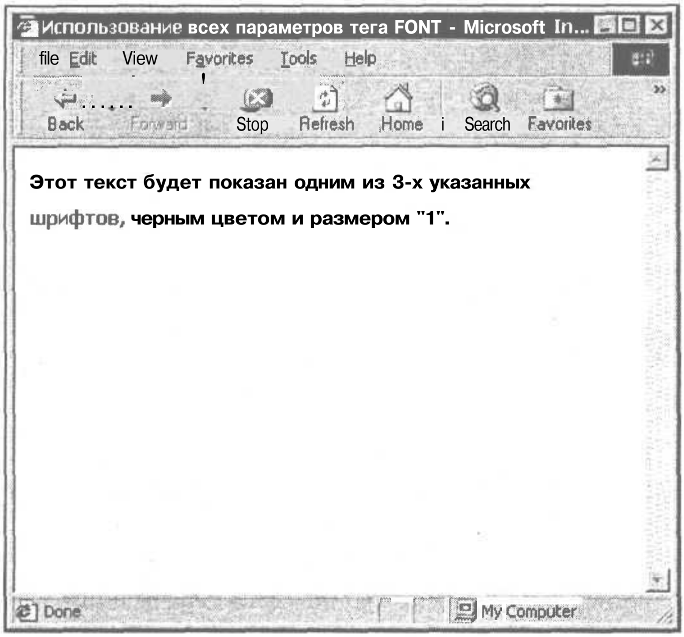


Рис. 2.6. Использование всех параметров тега <FONT>

Тег <I> — тег, аналогичный тегам логического форматирования <CITE> и <EM> (выделяет нужную часть текста курсивом). В ряде случаев использование <I> рекомендуется заменять соответствующими тегами логического форматирования.

Тег <SMALL> — предназначен для установки размера шрифта немного меньшего, чем остальные слова (поместив нужный фрагмент между парным тегом <SMALL>, получим эффект, противоположный действию тега <BIG>).

Тег **<SPAN>** — применяется в случае, когда выделенному фрагменту текста необходимо присвоить определенные свойства, а к помощи никакого логического тега прибегнуть нельзя.

Теги <s> и **<STRIKE>** — в последней версии спецификации HTML эти два тега были названы отмененными. Вместо них рекомендуется использовать тег <DEL>, также перечеркивающий выбранный фрагмент текста.

Тег **<SUB>** -- позиционирует фрагмент текста относительно нижней линии строки. Весьма удобный инструмент для написания математических и химических формул.

Тег **<SUP>** — размещает текстовый фрагмент относительно верхней линии строки. Также подходит для включения в разнообразные формулы (рис. 2.7).



Рис. 2.7. Использование тегов нижнего и верхнего индексов <SUB> и <SUP>

Тег <тт> — спецификация определяет его как тег для отображения телетайпа или текста, набранного моноширинным шрифтом. В ряде случаев может быть заменен тегом <CODE>.

Тег <U> — делает текст подчеркнутым.

В завершение разговора о тегах физического форматирования необходимо добавить, что разные теги могут использоваться одновременно в отношении одного текстового фрагмента. Например, часть текста и даже отдельного слова может быть написана и курсивом, и жирным начертанием, да еще подчеркнута и т. д. (листинг 2.7). Единственное, о чем следует помнить, это сохранение последовательности указания закрывающих тегов. Другими словами, не рекомендуется, открыв фрагмент текста тегами <в> и <FONT>, закрывать HTML-конструкцию в том же порядке (т. е. порядок закрывающих тегов должен быть обратным).

Листинг 2.7. Одновременное использование нескольких тегов физического форматирования

```
<HTML>
  <HEAD>
    <TITLE>Одновременное использование нескольких тегов физического фор-
матирования</TITLE>
```

```
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<FONT FACE="Verdana", "Arial" COLOR="gray" SIZE="3">В этом тексте исполь-
зовано <U>три различных тега</U> <B>физического форматирова-
ния.</B></FONT>
</BODY>
</HTML>
```

Полный перечень рассмотренных тегов физического форматирования приведен в табл. 2.2.

Таблица 2.2. *Теги физического форматирования*

Тег	Функция
в	Выделение жирным начертанием
BASEFONT	Определение свойств шрифта для всего документа
BIG	Выделение фрагмента текста большим размером, чем основной текст
BLINK	Создание эффекта мигания текста
FONT	Указание свойств шрифта (цвет, размер, название шрифта)
I	Выделение курсивом
SMALL	Выделение фрагмента текста меньшим размером, чем основной текст
SPAN	Присвоение свойств фрагменту текста при невозможности использования логических тегов
s, STRIKE	Обозначение удаленной информации
SUB	Создание нижнего индекса
SUP	Создание верхнего индекса
тт	Имитация телетайпа или моноширинного шрифта
U	Выделение подчеркиванием

Структурное форматирование

Как известно, любой текст имеет свою структуру. Книги разделены на части, главы и разделы. Газеты и журналы имеют отдельные рубрики и подзаголовки, которые, в свою очередь, включают фрагменты текста, также имеющие свою собственную внутреннюю структуру — абзацы, отступы, параграфы и пр.

Текст, размещенный в HTML-документе, не исключение, он также должен иметь логичную, понятную каждому пользователю структуру. Ведь от того, насколько просто и удобно будет восприниматься текст на Web-странице, зависит многое, прежде всего то, какое впечатление о HTML-документе сложится у посетителя.

Структурное форматирование в HTML подразумевает разбиение текстовых фрагментов электронного документа на логические блоки с информацией, которым соответствует определенный формат: абзац, текстовый блок, центрирование, отступы и перенос строки, горизонтальный разделитель, предварительно отформатированный текст и комментарии.

Рассмотрим каждый из названных элементов структурного форматирования.

Абзац

При наборе текста в каком-нибудь текстовом редакторе (Microsoft Word, WordPerfect и др.) для обозначения абзаца мы используем клавишу <Enter>. Такое действие дает программе команду обособить один фрагмент текста от другого, задав "красную строку". При создании HTML-документа для обозначения абзаца используется специальный тег <p>, который разделяет фрагменты текста вертикальным отступом (листинг 2.8). Простой перевод строки в данном случае не поможет: браузер, интерпретируя код, не воспримет отступ как команду создания абзаца и при выводе содержания на экран монитора объединит фрагменты текста вместе (рис. 2.8).

Листинг 2.8. Результат двойного перевода строки без использования тега <p>

```
<HTML>
  <HEAD>
    <TITLE>Результат перевода строки без использования тега P</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<FONT FACE="Tahoma" COLOR="#000000" SIZE="3">При создании HTML-документа
для обозначения абзаца используется специальный тег <CODE>P</CODE>, кото-
рый разделяет фрагменты текста вертикальным отступом.

Простой перевод строки в данном случае не поможет: браузер, интерпретируя
код, не воспримет отступ как команду создания абзаца и при выводе содер-
жания на экран монитора объединит фрагменты текста вместе.</FONT>
</BODY>
</HTML>
```

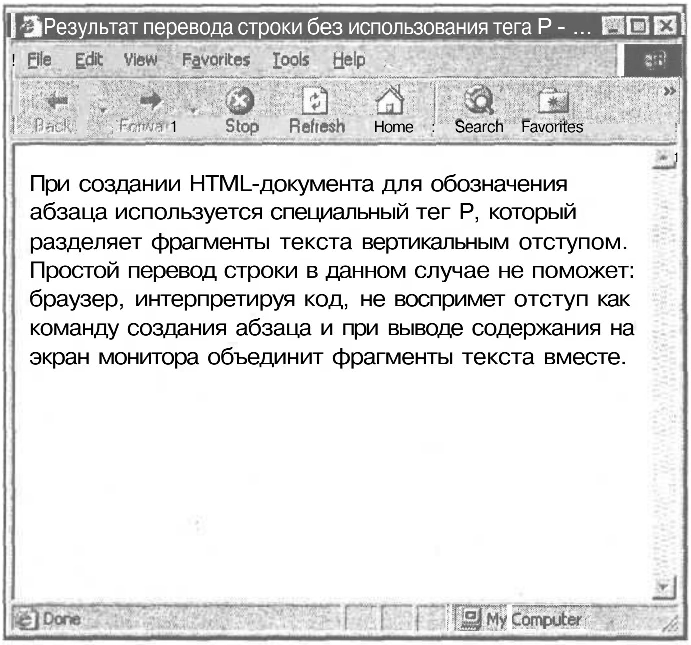


Рис. 2.8. Результат перевода строки без использования тега <P>

Тег <P> является парным, однако наличие закрывающего тега, согласно последней спецификации HTML, не обязательно. Так как <p> относится к разряду структурных тегов, включать другие элементы структурного форматирования он не может. Внутри него могут находиться только теги форматирования текста (логические и физические).

Тег <P> может содержать параметр ALIGN, отвечающий за тип горизонтального выравнивания текста в окне браузера (листинг 2.9):

- `ALIGN="left"` — текст выровнен по левому краю (значение параметра, принятое по умолчанию). При необходимости такого типа выравнивания указание параметра ALIGN не обязательно;
- `ALIGN="center"` — текст располагается посередине окна браузера. Использование данного значения не рекомендуется при работе с большими фрагментами текста, т. к. из-за различной длины слов восприятие абзаца затрудняется;
- `ALIGN="right"` — выравнивание текста по правому краю. Идеально подходит для создания эпиграфов, подписей, заголовков и пр. Использование при работе с большими текстовыми фрагментами нежелательно;
- `ALIGN="justify"` — выравнивание по ширине окна браузера. Значение `JUSTIFY` стало поддерживаться браузерами сравнительно недавно, однако во многих случаях именно этот тип выравнивания подходит для работы с текстом (рис. 2.9).

Листинг 2.9. Использование различных типов выравнивания

```
<HTML>
  <HEAD>
    <TITLE>Использование различных типов выравнивания</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<P>При создании HTML-документа для обозначения абзаца используется специ-
альный тег <CODE>P</CODE>, который разделяет фрагменты текста вертикаль-
ным отступом.</P>
<P ALIGN="center">Такой тип выравнивания подходит для заголовков докумен-
тов.</P>
<P ALIGN="right">Двойной перевод строки в данном случае не поможет: брау-
зер, интерпретируя код, не воспримет двойной отступ как команду создания
абзаца.</P>
<P ALIGN="justify">Значение JUSTIFY стало поддерживаться браузерами срав-
нительно недавно, однако во многих случаях именно этот тип выравнивания
подходит для работы с текстом.</P>
</BODY>
</HTML>
```

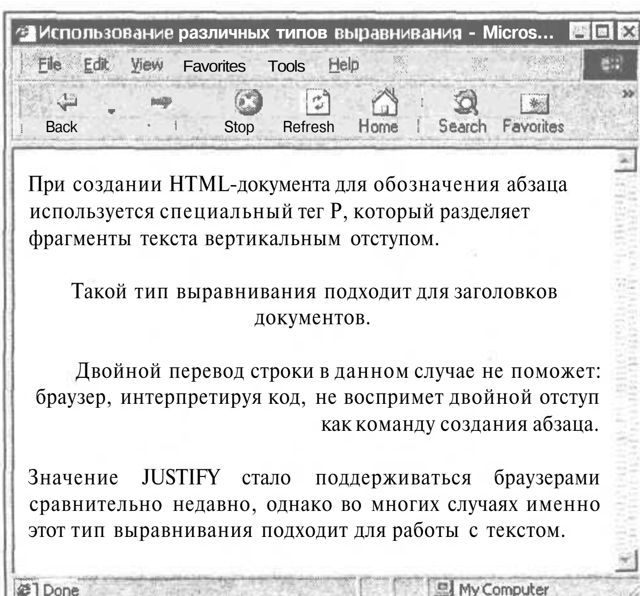


Рис. 2.9. Использование различных типов выравнивания

Как уже было сказано ранее, тег создания абзаца <p> позволяет использовать все теги форматирования текста — как логического, так и физического. Следующий пример (листинг 2.10) иллюстрирует эту возможность (рис. 2.10).

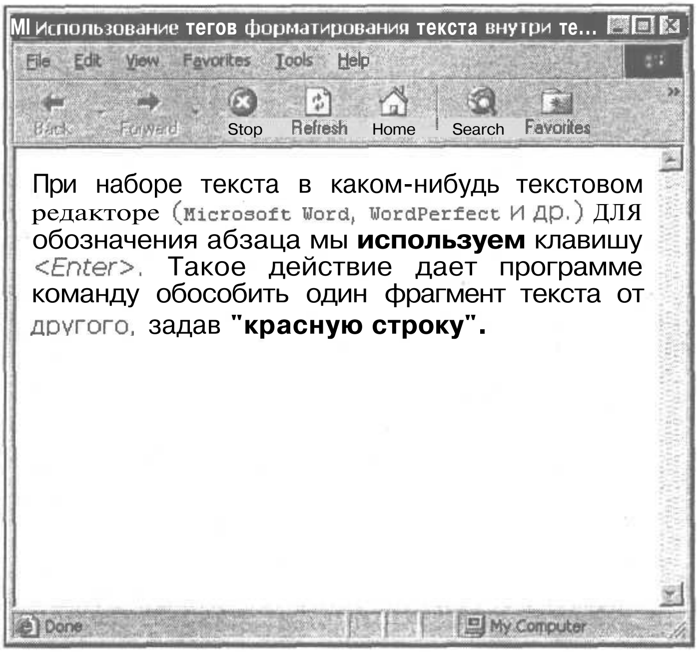


Рис. 2.10. Использование тегов форматирования текста внутри тега <P>

Листинг 2.10. Использование тегов форматирования текста внутри тега <P>

```
<HTML>
  <HEAD>
    <TITLE>Использование тегов форматирования текста внутри тега
P</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<P ALIGN="justify"><FONT FACE="Verdana" SIZE="3">
При наборе текста в каком-нибудь текстовом редакторе (<TT>Microsoft
Word</TT>, <TT>WordPerfect</TT> и др.) для обозначения абзаца мы
<STRONG>используем</STRONG> клавишу <EM>&lt;Enter&gt;</EM>. Такое дейст-
```

```
вие дает программе команду обособить <U>один фрагмент текста от друго-
го</U>, задав <FONT COLOR="red"><BIG>"красную строку"</BIG></FONT>.
</FONT></P>
</BODY>
</HTML>
```

В данном примере в конце кода стоят два закрывающих тега </FONT>: один завершает действие свойства COLOR=red, другой — закрывает правила отображения свойств типа и размера шрифта (FONT FACE="Verdana" SIZE="3").

Текстовый блок

В случае необходимости указания специальных свойств отдельному фрагменту текста используются теги текстовых блоков <DIV> и <SPAN>. Изменение свойств осуществляется посредством назначения выбранному фрагменту текста стиля CSS, например:

```
<DIV STYLE="COLOR: GRAY;">
Фрагмент текста, набранный серым цветом
</DIV>
```

Более подробно о назначении стилей будет рассказано в *главе 9.*

Однако следует помнить, что между <DIV> и <SPAN> имеются существенные отличия. Во-первых, <DIV> является исключительно структурным тегом, а <SPAN> берет начало в области физического форматирования текста. Во-вторых, <DIV> создает принудительный перенос строки на одну позицию после своего закрывающего тега (в отличие от тега абзаца <P>, который осуществляет перенос на две позиции), поэтому задавать с его помощью отдельные свойства фрагмента внутри абзаца нельзя — это вызовет принудительный перенос строки. Если такая необходимость возникла, лучше использовать тег <SPAN>, который позволяет назначать новые правила отображения текстовых фрагментов без изменения структуры документа.

Для обоих элементов обязательно наличие закрывающих тегов, а также возможно добавление параметра типа выравнивания ALIGN, например (рис. 2.11):

```
<P ALIGN="justify">
Начало основного текста документа. <DIV ALIGN="right">Фрагмент тек-
ста.</DIV> Окончание основного текста документа.
</P>
```

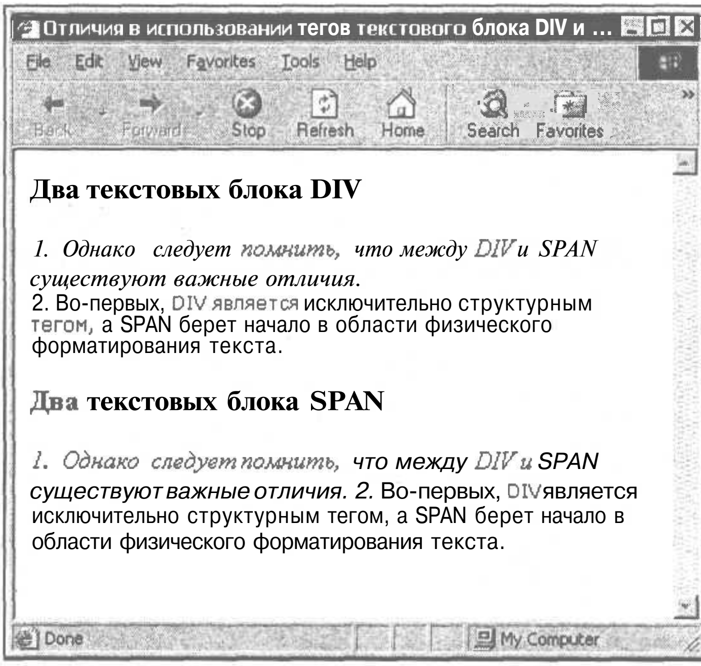


Рис. 2.11. Отличия в использовании тегов текстового блока <DIV> и <SPAN>

Центрирование

Центрирование любых элементов HTML-документа может быть осуществлено с помощью тега <CENTER>. Все данные, размещенные внутри тега-контейнера <CENTERX/CENTER>, подлежат горизонтальному выравниванию по середине окна браузера. В сущности, тег <CENTER> представляет собой аналог значения `ALIGN="center"` того тега, результат отображения которого будет отцентрирован на экране монитора. К примеру, результат отображения фрагмента кода таблицы

```
<CENTER>
<TABLE><TR><TD></TD></TR></TABLE>
</CENTER>
```

и фрагмента кода

```
<TABLE ALIGN="center"><TR><TD></TD></TR></TABLE>
```

будет абсолютно одинаков. В первом случае использован структурный тег центрирования, во втором — указан соответствующий тип выравнивания всей таблицы относительно окна браузера.

Отступы и перенос строки

Часто во многих печатных и книжных изданиях мы можем наблюдать, что начало новой строки абзаца начинается после небольшого отступа. В текстовом редакторе эта возможность регулируется клавишей табуляции <Tab> или перемещением ползунка линейки настроек рабочей области документа (рис. 2.12).

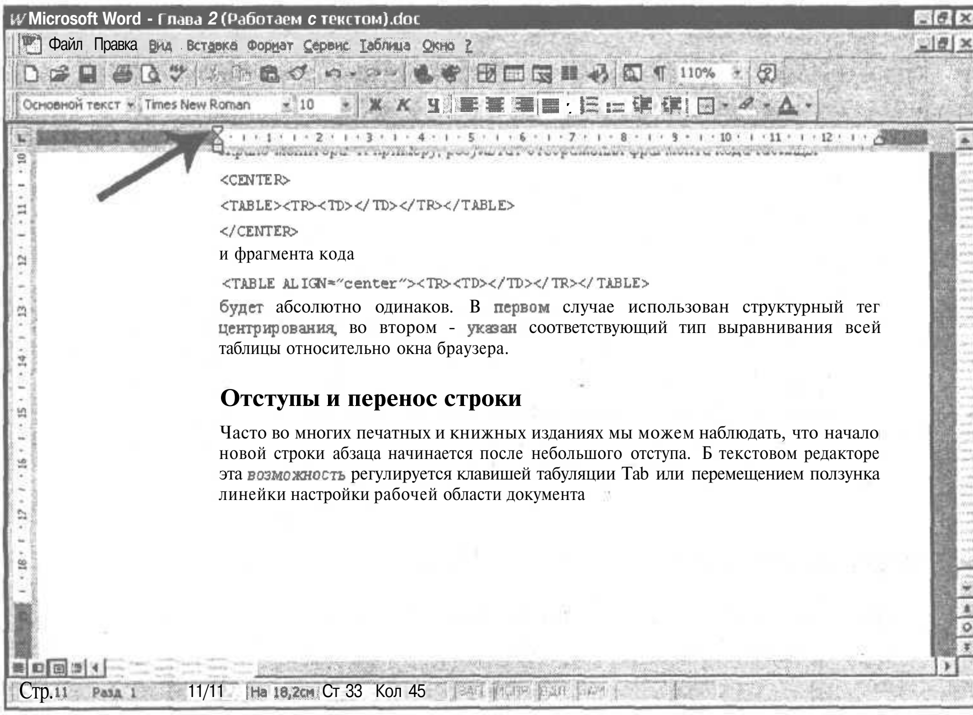


Рис. 2.12. Ползунок линейки рабочей области в текстовом редакторе Microsoft Word

В HTML размер отступа определяется кодовой конструкцией ` `, которая визуально представляет собой обычный пробел (листинг 2.11). Создавая текст в документе, не обязательно между каждым словом вставлять ` ` — браузер и так поймет, что нужно сделать отступ. Но если вам потребуется сделать отступ большего размера и вы прямо в коде создадите несколько пробелов, браузер интерпретирует такие пропуски в виде единственного пробела (рис. 2.13).

Листинг 2.11. Создание отступов с помощью конструкции finbsp;

```
<HTML>
   <HEAD>
```

```
    <TITLE>Создание отступов</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<P ALIGN="justify"><FONT FACE="Verdana" SIZE="3">
    При наборе текста в каком-нибудь текстовом редак-
торе (Microsoft Word, WordPerfect и др.) для обозначения абзаца мы ис-
пользуем клавишу &lt;Enter&gt;.
</FONT></P>
<P ALIGN="justify"><FONT FACE="Verdana" SIZE="3">
    Такое действие дает программе команду обособить
один фрагмент текста от другого, задав "красную строку".
</FONT></P>
</BODY>
</HTML>
```

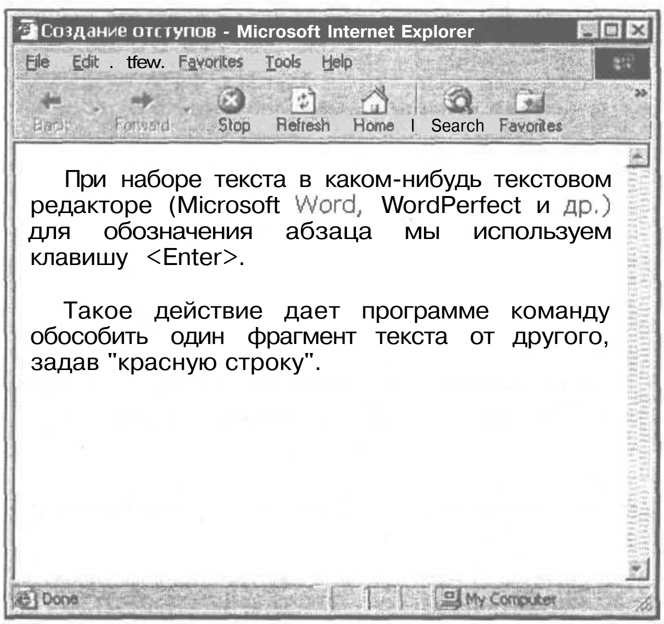


Рис. 2.13. Создание отступов с помощью конструкции

Конструкция   относится к группе специальных символов HTML, о которых пойдет речь в *разд. "Специальные символы" этой главы.*

Размер экрана монитора ограничен, а ширина рабочего окна браузера имеет свои рамки, вынуждающие переносить слова на следующую строку. Браузер переносит слова после пробела или некоторых символов (дефис, прямой

слэш и пр.). То есть разделить целое слово, разместив начало на одной строке, а окончание — на другой, браузер сам не может. Однако в ходе создания HTML-документа может возникнуть необходимость переноса строки (например, при размещении отрывка стихотворения каждая фраза должна начинаться с новой строки). Если мы переместим строки стихотворения посредством клавиши <Enter>, браузер отобразит отрывок в виде одного предложения. В таких случаях нужно использовать тег принудительного переноса строки `<BR>` (рис. 2.14, листинг 2.12).

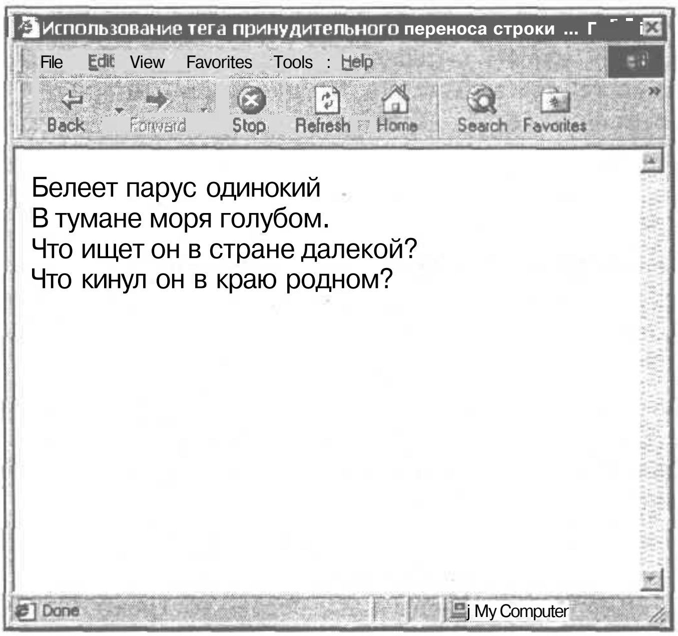


Рис. 2.14. Использование тега принудительного переноса строки

Листинг 2.12. Использование тега принудительного переноса строки `<BR>`

```
<HTML>
  <HEAD>
    <TITLE>Использование тега принудительного переноса строки BR</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#C0FF00" ALINK="#C0FFC0"
VLINK="blue">
<PXFONT FACE="Arial" SIZE="4">
Белеет парус одинокий<BR>
В тумане моря голубом.<BR>
```

```
Что ищет он в стране далекой?<BR>
Что кинул он в краю родном?
</FONTX/P>
</BODY>
</HTML>
```

Для переноса на конкретное число строк тег
 ставится нужное количество раз.

Также у разработчика HTML-документа может возникнуть ситуация, противоположная только что описанной, — необходимость запрета переноса строки. Например, слово "интернет-каталог" при определенных размерах окна браузера может быть разделено на две части: "интернет-" и "каталог", что мешает восприятию слова целиком.

Во избежание такой ситуации слово или словосочетание, для которого следует запретить перенос строки, необходимо поместить внутри тега-контейнера <NOBRX/NOBR>.

```
<P><FONT COLOR="#003366"><STRONG>
Одним из наиболее эффективных способов рекламы Web-сайта является регистрация в <NOBR>интернет-каталогах</NOBR> и поисковых системах.
</STRONG></FONT></P>
```

В этом случае переноса не произойдет, но зато возможны такие нежелательные последствия, как нарушение типа выравнивания текста, внезапный перенос смежных строк и возникновение необходимости горизонтальной прокрутки окна браузера.

Эти негативные последствия можно частично предотвратить с помощью тега <WBR>. Он размещается в том месте слова, которое является (по мнению автора документа) наименее опасным для переноса строки. Однако применение тега <WBR> сильно затруднено тем, что многие браузеры (в частности, Netscape) его не поддерживают, а Internet Explorer распознает <WBR> только при наличии в коде тегов <NOBR>.

Горизонтальный разделитель

Ранее мы говорили об элементах структурного форматирования, которые разделяли фрагменты текста логически — посредством соответствующей HTML-разметки. В этом смысле использование горизонтальных разделителей (или просто линий) можно назвать своеобразным исключением из правил, ибо они проводят визуальную границу между различными текстовыми фрагментами (рис. 2.15).

Горизонтальный разделитель описывается тегом <HR> (закрывающего тега нет) и имеет целый ряд параметров, позволяющих изменять визуальные свойства линии (добавлять рельефность и цвет, регулировать длину, задавать

тип выравнивания и пр.). Перечень параметров тега `<HR>` приведен в табл. 2.3.

Таблица 2.3. *Параметры тега* `<HR>`

Параметр	Функция
WIDTH	Указание длины линии (в процентах от ширины окна браузера или пикселах)
SIZE	Указание толщины линии (в пикселах)
ALIGN	Обозначение типа выравнивания линии (LEFT, CENTER, RIGHT)
COLOR	Выбор цвета линии (название, шестнадцатеричный код или формат RGB)
NOSHADE	Отмена эффекта рельефности линии

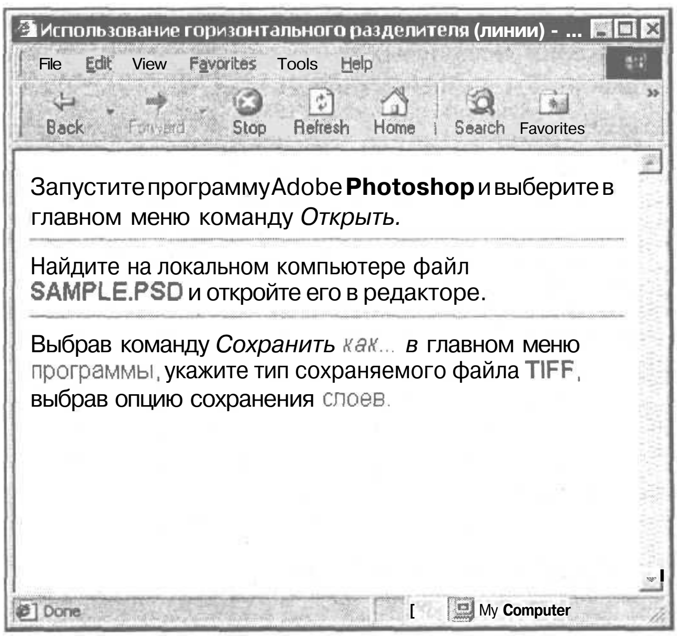


Рис. 2.15. Использование горизонтального разделителя (линии)

Пример HTML-конструкции для создания горизонтальной линии:

```
<HR WIDTH="90%" SIZE="1" ALIGN="CENTER" COLOR="#000000" NOSHADE>
```

В заключение следует добавить, что с точки зрения форматирования структуры документа, горизонтальная линия выполняет не только функцию визуального разделителя отдельных фрагментов текста, но и подразумевает перенос строки.

Предварительно отформатированный текст

Тег-контейнер предварительно отформатированного текста <PRE></PRE> позволяет выводить на экран монитора текст такого формата, каким он представлен в окне текстового редактора (со всеми переносами строк, отступами, абзацами и прочим форматированием). Такая возможность весьма удобна, например, при отображении фрагментов программного кода "как есть", без его интерпретации браузером.

Визуально результат действия тега <PRE> выглядит как текст, набранный моноширинным шрифтом. В спецификации существует еще несколько похожих по функции тегов (<PLAINTEXT>, <XMP> и <LISTING>), однако они признаны устаревшими и Консорциум W3C рекомендует использовать тег <PRE>.

Комментарии

Если горизонтальная линия является визуальным разделителем фрагментов текста, то комментарии невидимы на экране монитора, но *в* то же время могут нести большой объем информации.

Любой HTML-комментарий должен начинаться с конструкции `<!--` и заканчиваться -->. Между ними может находиться любой текст, цифры, символы и прочее, за исключением тегов.

Для чего же нужны HTML-комментарии? Во-первых, они позволяют размечать структуру HTML-кода, давая заголовки различным логическим блокам элементов. Впоследствии такая разметка поможет быстро сориентироваться и найти необходимый фрагмент кода.

```
<!-- Начало блока новостей -->
Код блока новостей...
<!-- Окончание блока новостей -->
```

Во-вторых, в комментариях можно указать информацию об авторском праве и прочие персональные данные, которые будут проиндексированы поисковыми системами.

```
<!-- © 2002-2003. Алексей Петюшкин. Все права защищены -->
```

При всем удобстве включения в HTML-код комментариев необходимо помнить, что просмотреть код электронного документа может практически любой пользователь Интернета, открыв нужную Web-страницу в любом текстовом редакторе. Поэтому размещать в комментариях информацию конфиденциального, коммерческого или личного характера не рекомендуется.

Перечень всех рассмотренных элементов и тегов структурного форматирования приведен в табл. 2.4.

Таблица 2.4. *Элементы и теги структурного форматирования*

Элемент	Функция	Используемые теги
Абзац	Создание текстовых абзацев	<P>
Текстовый блок	Визуальное выделение отдельных фрагментов текста	<DIV> <SPAN>
Центрирование	Горизонтальное выравнивание элементов по ширине окна браузера	<CENTER>
Отступы и перенос строки	Создание отступов в тексте и создание/запрет переносов строки	 <NOBR> <WBR>
Горизонтальный разделитель	Визуальное разделение фрагментов текста с помощью линии	<HR>
Предварительно отформатированный текст	Вывод отформатированного в текстовом редакторе фрагмента на экран монитора в формате "как есть"	<PRE> <PLAINTEXT> <XMP> <LISTING>
Комментарии	Включение дополнительной информации о документе в HTML-код, невидимой при просмотре в браузере	<!-- -->

Специальные символы

К специальным символам HTML относятся символы, не входящие в состав стандартных ASCII-кодов. Их реализация в HTML-документах возможна при помощи отдельных кодовых конструкций или числовых комбинаций.

Специальные символы используются во многих интернет-ресурсах. Это знак авторского права и зарегистрированной торговой марки, значки иностранных валют и математические символы, дробные числа и элементы HTML-форматирования, наконец, буквы иностранных алфавитов и многое другое. Например, значок доллара ($) можно легко ввести с помощью соответствующей клавиши, не опасаясь за корректность отображения данного символа в любых кодировках, моделях и версиях браузеров. А вот отобразить таким же образом значок английской денежной единицы — фунта стерлингов — вряд ли удастся.

Условно все специальные символы HTML можно разделить на три большие категории:

- символы, отображающие элементы HTML-форматирования;
- символы оформления документа;
- буквы иностранных алфавитов.

Рассмотрим более подробно каждую категорию.

Символы, отображающие элементы HTML-форматирования

В ходе создания того или иного HTML-документа часто бывает необходимо отобразить в браузере символы, используемые в спецификации языка HTML: угловые скобки, обозначающие теги (< >), знак амперсанда (&) и т. д. Чаще всего это случается при добавлении в исходный документ фрагмента HTML-кода, который впоследствии должен отображаться в окне браузера именно как кодовый листинг, а не результат его исполнения.

Например, вы хотите обменяться с другими Web-сайтами кнопками, а для удобства обмена вам нужно предоставить посетителям вашего интернет-ресурса готовый HTML-код, который останется только выделить (прямо в окне браузера), скопировать в буфер обмена и вставить в соответствующее место на Web-странице:

```
<A HREF="http://www.site.ru/"><IMG
SRC="http://www.site.ru/images/button.gif" WIDTH="88" HEIGHT="31"
BORDER="0"></A>
```

Если поместить этот код в данном виде на Web-страницу, то браузер загрузит указанный в коде графический файл со ссылкой на установленный в теге <A> ресурс. Чтобы в браузере отображался сам код, нужно все левые угловые скобки в коде заменить на <, а правые — на >. После замены всех символов специальными кодовыми конструкциями первоначальный фрагмент кода примет следующий вид:

```
&lt;A HREF="http://www.site.ru/"&gt;&lt; IMG
SRC="http://site.ru/images/button.gif" WIDTH="88" HEIGHT="31"
BORDER="0"&gt;&lt;/a&gt;
```

Символы оформления документа

К так называемым символам оформления HTML-документов относятся значки авторского права, зарегистрированной торговой марки, символы иностранных валют, математические обозначения ("плюс-минус", "умножить"

и др.), а также прочие символы. Приведенный на рис. 2.16 пример может быть получен за счет вставки в код документа следующих конструкций:

`£` — Фунт стерлингов

`®` — Зарегистрированная торговая марка

`±` — Знак "плюс-минус"

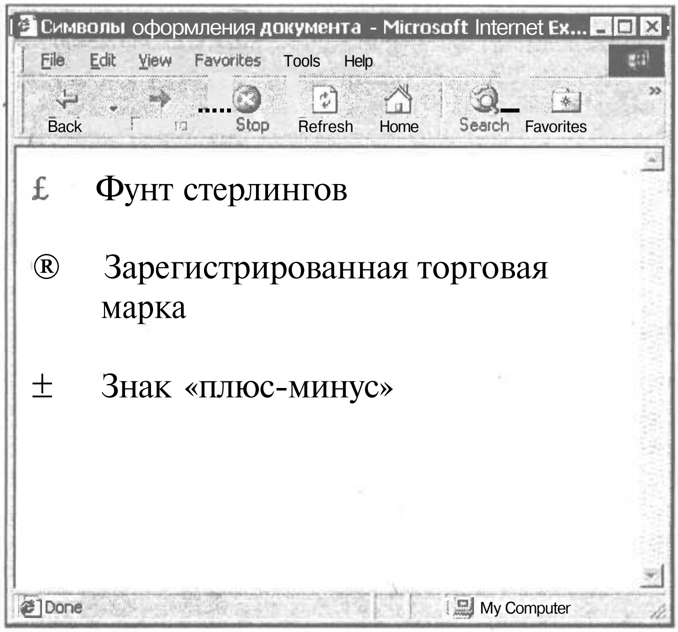


Рис. 2.16. Символы оформления документа

Как видно, первые два символа выводятся браузером с помощью специальной кодовой конструкции (знак амперсанда (&) и соответствующее наименование), а последний — путем указания амперсанда (&), знака диеза (#) и числового кода.

Данный пример не означает, что значок английской национальной валюты и зарегистрированной марки можно получить только посредством кодовой конструкции, а знак "плюс-минус" — посредством числового обозначения. Для каждого специального символа (будь это символ оформления документа, буква иностранного алфавита или символ отображения элементов HTML) существуют два способа записи — кодовый и числовой. К примеру, конструкции `§` и `§` совершенно одинаково будут выводиться в окне браузера в виде значка параграфа (§).

Зная конструкции символов оформления, можно существенно облегчить процесс создания HTML-документов. Например, если вы создаете HTML-код

в Блокноте (NotePad, входящий в состав ОС Windows) и вам обязательно нужно использовать угловые кавычки («»), совсем не обязательно заходить в текстовый редактор типа Microsoft Word, вводить желаемый тип кавычек (очень часто Word по умолчанию отображает верхние кавычки), копировать их и, снова открыв окно Блокнота, вставлять в документ. Надо просто указать кодовую или числовую конструкцию для нужного типа кавычек и браузер отобразит символ необходимым образом.

Если вы хотите разместить в HTML-документе некоторые математические данные, то внешний вид уравнения, содержащего в качестве знака умножения букву "x", сильно уступает уравнению, где применяется специальная конструкция знака "умножить" (×).

Следует сказать, что описываемые конструкции специальных символов не используются в отображении знака степени и числа атомов в химическом соединении. Для этого (а также для иных целей) предназначены теги-контейнеры (верхний индекс) и (нижний индекс).

Буквы иностранных алфавитов

Когда создается HTML-документ на русском или английском языке, никаких затруднений с отображением букв в браузере, разумеется, возникнуть не может. Все они есть на раскладке клавиатуры компьютера. Представим ситуацию, в которой необходимо разработать электронный документ, к примеру, на португальском языке, алфавит которого содержит некоторые буквы, чье начертание характерно только лишь для данного языка. Как можно выйти из такого положения?

Существует несколько вариантов. Первый заключается в том, чтобы добавить в раскладку клавиатуры новый язык, включить в используемом текстовом редакторе поддержку португальского и, выбрав на Панели задач значок подключения языка, спокойно набирать текст. Спокойно ли? Мы забыли о конечной цели ввода такого текста — быть интерпретированным, причем корректно, браузером.

Если же, набрав предложения на португальском в текстовом редакторе, скопировать их и перенести в Блокнот, то этот текст отобразится без специальных символов, свойственных данному языку. На рис. 2.17 показаны некоторые особенности португальского языка: буквы "ç" и "ã" пишутся с использованием специальных элементов вверху и внизу (они называются "седиль" и "тильда", соответственно), чего никак нельзя достигнуть путем простого переноса фрагмента из текстового редактора в программу создания HTML-документов, подобную Блокноту.

Второй вариант - прибегнуть к услугам визуальных Web-редакторов (FrontPage, Word 2000 и др.), которые включают мультиязыковую поддержку. Однако последующая программная конвертация исходного текста

в формат HTML может привести, во-первых, к некорректному отображению его на Web-странице некоторыми моделями и версиями браузеров, а во-вторых — к генерации совершенно ненужного и бесполезного кода, только увеличивающего конечный размер файла.

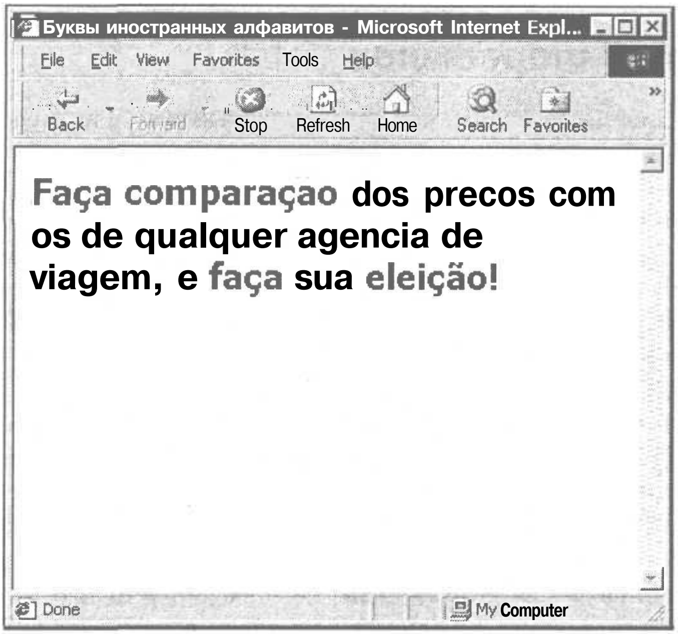


Рис. 2.17. Буквы иностранных алфавитов

Третий вариант подразумевает использование специальных кодовых или числовых конструкций, предназначенных для корректной визуализации нестандартных букв иностранного алфавита. Например, чтобы браузер правильно отобразил последнее слово в тексте на португальском языке, показанном на рис. 2.17, в код документа нужно вписать следующую конструкцию:

```
elei&ccedil;&atilde;o
```

В результате — никаких проблем с кросс-браузерной совместимостью и "утяжеленным" размером HTML-файла. К тому же люди, знающие португальский язык, не встретят в таком документе непонятных слов и выражений.

В завершение разговора о специальных символах, следует упомянуть один важный момент, который прежде всего касается случаев использования букв иностранных алфавитов. В мета-определениях таких HTML-документов необходимо указывать либо родную (национальную) кодировку, либо кодировку ISO8859, иначе некоторые браузеры (например, Netscape) вместо

желанных специальных символов могут отобразить смесь непонятных знаков. Internet Explorer, в отличие от своего конкурента на рынке браузеров, неприхотлив и корректно отображает все специальные символы даже в русских кодировках (полный перечень всех специальных символов, а также их кодовые и числовые конструкции можно найти в *приложении 1*).

Создание гиперссылок

Читая книги, мы часто видим, как автор, раскрывая ту или иную тему, ссылается на другой раздел или вообще на совершенно другой печатный источник. Вот и я, завершив рассказ о специальных символах HTML, упомянул *приложение 1,* в котором можно найти некую дополнительную информацию.

В Интернете размещены миллионы электронных документов, часто похожих по тематике и ориентированных на одну и ту же пользовательскую аудиторию. Переход на другие документы осуществляется только благодаря возможности связывания одного HTML-документа с другим с помощью гипертекстовых ссылок.

По сути, любая гипертекстовая ссылка — это указатель адреса в глобальной сети, по которому можно моментально перейти из окна браузера. Однако успешный переход по ссылке возможен в двух случаях — если ресурс, на который ссылается документ, существует и если структура гиперссылки верна с точки зрения HTML.

Первый фактор является объективным, не зависящим от нас, т. к. разработчик электронного документа, однажды поставив в нем ссылку на внешний ресурс, может и не знать о том, что этот ресурс прекратил свое существование, или же переместился на другой адрес, или же временно закрыт и пр.

Второй фактор относится к разряду субъективных, ибо только от создателя HTML-документа может зависеть, сумеет ли посетитель перейти по ссылке или она составлена неверно. Чтобы последнего не произошло, рассмотрим структуру и правила описания гипертекстовых ссылок.

Структура гиперссылок

Любая гиперссылка состоит из двух важных частей: указателя ссылки ("якоря" — от англ. "anchor") и адреса ресурса, на который необходимо осуществить переход. Внешне отличить гиперссылку от обычного текста очень просто: при наведении курсора мыши на ссылку указатель принимает вид руки с указательным пальцем, как бы показывающим, что этот текст содержит гиперссылку. Сама ссылка подчеркивается (в случае если указателем является текст).

В качестве указателя ссылки может выступать текст (отдельное слово, фразы и даже целые страницы текста) и графические изображения. В ряде случаев возможно объединение графики и текста в рамках единого указателя ссылки.

Указатель ссылки описывается тегом <A>, а адрес перехода реализован с помощью параметра HREF, значением которого является путь к тому или иному интернет-ресурсу (закрывающий тег </A> является обязательным). Пример простейшей гиперссылки приведен соответственно на рис. 2.18 и в листинге 2.13.

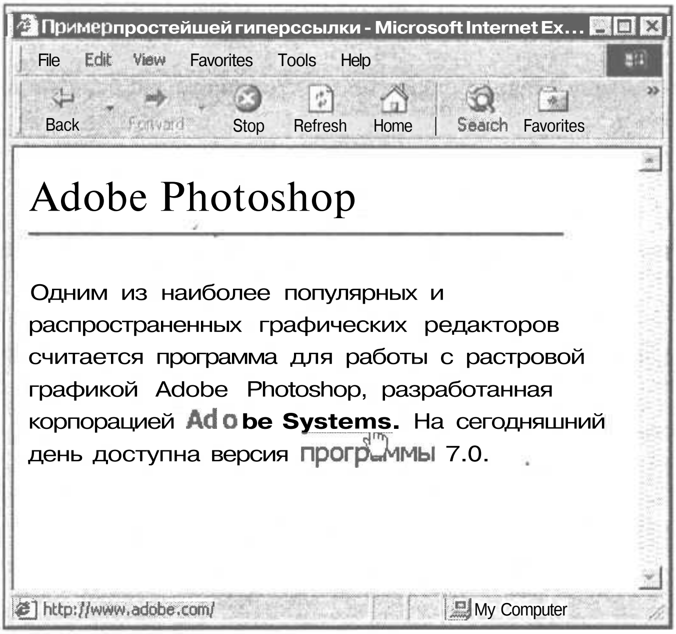


Рис. 2.18. Пример простейшей гиперссылки

Листинг 2.13. Пример простейшей гиперссылки

```
<HTML>
  <HEAD>
    <TITLE>Пример простейшей гиперссылки</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#FF0000"
VLINK="blue">
<FONT COLOR="gray" SIZE="6">Adobe Photoshop</FONTXHR COLOR="#003366"
WIDTH="90%" ALIGN="left">
<PXFONT FACE="Tahoma", "Verdana" SIZE="4">
Одним из наиболее популярных и распространенных графических редакторов
считается программа для работы с растровой графикой Adobe Photoshop, раз-
работанная корпорацией <A HREF="http://www.adobe.com/"><B>Adobe
Systems</B></A>. На сегодняшний день доступна версия программы 7.0.
```

```
</FONTX/P>
</BODY>
</HTML>
```

В случае когда указателем ссылки является графическое изображение, внутри тега-контейнера <A> указывается путь к рисунку, выполняющего роль гипертекстовой ссылки (листинг 2.14).

Листинг 2.14. Пример гиперссылки с использованием графики в качестве указателя

```
<HTML>
  <HEAD>
    <TITLE>Пример гиперссылки с использованием графики в качестве указа-
теля</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#FF0000"
VLINK="blue">
<P ALIGN="center">
<A HREF="http://www.adobe.com/"><IMG
SRC="http://www.adobe.com/images/adobe.gif" WIDTH="24" HEIGHT="31"
ALT="Adobe Systems" BORDER="0"></A>
</P>
</BODY>
</HTML>
```

Следует обратить внимание на то, что значение параметра фаницы рисунка BORDER равно нулю, в противном случае браузер, интерпретируя изображение как указатель ссылки, поставит вокруг рисунка рамку (более подробно о правилах описания фафических изображений в HTML будет изложено в *главе 5)*.

Если необходимо объединить в структуре одной ссылки и текст, и графику, HTML-код принимает вид, представленный соответственно в листинге 2.15 и на рис. 2.19.

Листинг 2.15. Пример гиперссылки с использованием в качестве указателя и текста, и графики

```
<HTML>
  <HEAD>
    <TITLE>Пример гиперссылки с использованием в качестве указателя и
текста, и графики</TITLE>
```

```
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#FF0000"
VLINK="blue">
<FONT COLOR="gray" SIZE="6">Adobe Photoshop</FONT><HR COLOR="#003366"
WIDTH="90%" ALIGN="left">
<PXFONT FACE="Tahoma", "Verdana" SIZE="4">
Одним из наиболее популярных и распространенных графических редакторов
считается программа для работы с растровой графикой Adobe Photoshop, раз-
работанная корпорацией <A HREF="http: //www.adobe.com/"><IMG
SRC="http://www.adobe.com/images/adobe.gif" WIDTH="24" HEIGHT="31"
ALT="Adobe Systems" BORDER="0"><B>Adobe Systems</B></A>. На сегодняшний
день доступна версия программы 7.0.
</FONTX/P>
</BODY>
</HTML>
```

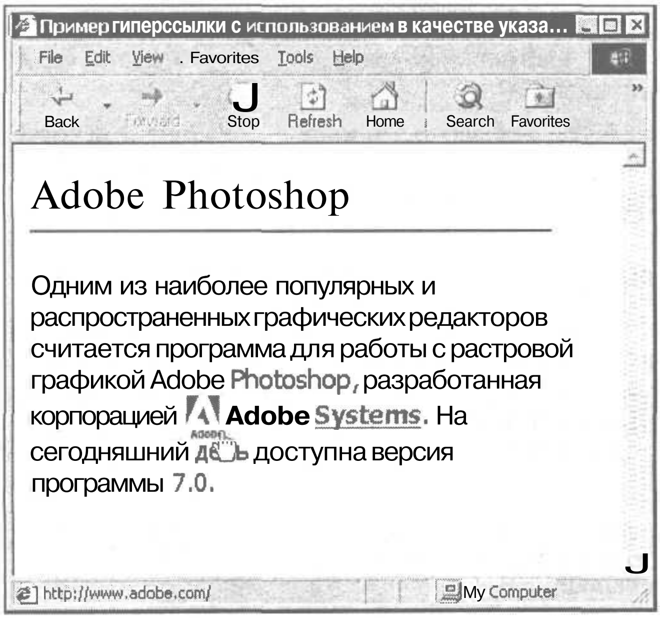


Рис. 2.19. Пример гиперссылки с использованием в качестве указателя и текста, и графики

Помимо HREF, тег <A> может содержать еще и другие параметры, например, STYLE, CLASS, NAME, TITLE и TARGET. Первые два параметра используются для создания стилевых шаблонов гиперссылки (о которых речь пойдет в *главе 9)*, третий параметр предназначен для указания имени внутренней ссылки (подробнее в *разд. "Правила описания гиперссылок" этой главы)*.

Параметр TITLE используется для создания всплывающей подсказки при наведении курсора мыши на гиперссылку. Подсказка может выступать в качестве комментария или описания к ссылке, ведущей на электронный документ:

```
<A HREF="http://www.site.ru/" TITLE="Описание документа">Ссылка на документ</A>
```

Необходимо добавить, что параметр TITLE тега <A> поддерживается только браузером Internet Explorer.

Что касается параметра TARGET, то он определяет назначение перехода по ссылке. Например, нажав на гиперссылку, можно открыть документ в текущем или в отдельном (новом) окне. Это в ряде случаев является принципиальным моментом. Например, если вы не хотите уводить посетителя со своей Web-страницы, то стоит делать ссылки со следующей структурой:

```
<A HREF="http://www.site.ru/page.html" TARGET="_blank">Ссылка на документ</A>
```

Наличие знака нижнего подчеркивания перед значением blank не является обязательным атрибутом для большинства популярных браузеров, однако Консорциум W3C настоятельно рекомендует указывать полную конструкцию параметра TARGET.

В завершение следует добавить, что параметр TARGET и некоторые другие его значения (помимо `_blank`) используются в описании так называемой фреймовой структуры, о чем будет рассказано в *главе 6*.

Правила описания гиперссылок

Начну с того, что все гиперссылки можно разделить на два типа: внешние и внутренние. Внешние ссылки ведут на другие ресурсы глобальной сети или другие документы одного Web-сайта, а внутренние позволяют посетителю путешествовать в пределах одного HTML-документа.

Внешние ссылки

Рассмотрим правила описания внешних гипертекстовых ссылок

Путь адреса ссылки

Структура внешней ссылки состоит из указателя (якоря) и адреса. Существуют два способа записи адреса перехода по внешней ссылке.

- Абсолютный. В значении параметра HREF указывается полный путь к ресурсу, ссылка на который ставится в документе, например:

  ```
  <A HREF="http://www.site.ru/docs/page1.html">Ссылка на страницу 1</A>
  ```

  Независимо от того, с какого документа пользователь нажмет на такую ссылку, он перейдет по адресу `http://www.site.ru/docs/page1.html`.

- Относительный. В значении параметра HREF указывается конечный документ, относительно которого размещена страница, содержащая ссылку, например:

  <A HREF="page2.html">Ссылка на страницу 2 со страницы 1</A>

  Такой формат записи внешней ссылки подразумевает расположение файла page2.html (на который указывает гиперссылка) в том же каталоге, что и файл page1.html (с которого будет осуществляться переход). В нашем случае оба файла расположены в каталоге docs, находящемся на Web-сайте **www.site.ru**.

Теперь попытаемся усложнить ситуацию и представим, что файл page2.html размещен в корне сайта **www.site.ru**, а файл page1.html по-прежнему находится в каталоге docs. В этом случае запись внешней ссылки с использованием относительного пути к файлу примет следующий вид:

<A HREF="../page2.html">Ссылка на страницу 2 со страницы 1</A>

Таким образом, с помощью относительного пути перехода по ссылке мы велели браузеру подняться на два уровня вверх (минуя уровень файлов, размещенных в каталоге docs, и уровень размещения самого каталога docs).

Следует помнить, что внешняя ссылка, путь к которой прописан относительно, может вести только на документы текущего сайта. Перенести посетителя на посторонние интернет-ресурсы (например, с сайта **www.site1.ru** на **www.site2.ru**) такая ссылка не может.

Указание протокола перехода по ссылке

Понятие Интернета как глобальной информационной сети подразумевает не только World Wide Web (WWW, Всемирная Паутина), в состав которой входит большинство интернет-ресурсов. Интернет — это более емкая инфраструктура, включающая в себя различные информационные сервисы, работа которых реализуется с помощью так называемых протоколов — наборов технологических правил взаимодействия документов друг с другом.

Например, WWW работает на основе протокола HTTP (HyperText Transfer Protocol, протокол передачи гипертекста). Помимо этого, существуют такие технологии, как FTP (File Transfer Protocol, протокол передачи файлов), E-mail (служба электронной почты), News (группы новостей), Gopher и др.

Вполне возможна ситуация, когда разработчику HTML-документа понадобится поставить ссылку на другие, отличные от HTTP сервисы. В этом случае структура гиперссылки остается прежней — указатель (текстовый и/или графический) и адресная часть. Однако последняя может меняться в зависимости от выбранной технологии передачи данных.

Например, ссылка на адрес электронной почты, которая сегодня получила гораздо большее распространение, чем интернет-ресурсы, может выглядеть так:

<A HREF="mailto:**mail@site.ru**">Ссылка на адрес электронной почты</A>

Если пользователь нажмет на такую ссылку, на его компьютере запустится установленная по умолчанию программа чтения и отправления электронной почты (например, Microsoft Outlook, The Bat!, Netscape Messenger, Eudora и др.). В этом случае можно несколько усложнить структуру ссылки, добавив в нее готовое поле заголовка электронного письма:

```
<A HREF="mailto:mail@site.ru?Subject=Здравствуйте!">Ссылка на адрес элек-
тронной почты</A>
```

Такая гиперссылка не только запустит почтовую программу и откроет окно нового письма, но и добавит в поле заголовка электронного сообщения Subject обращение "Здравствуйте!".

Правила составления гиперссылок на различные информационные сервисы Интернета приведены в табл. 2.5.

Таблица 2.5. *Правила составления гиперссылок на различные информационные интернет-сервисы*

Название информационного сервиса	Пример гиперссылки
World Wide Web	`http://www.site.ru/`
E-mail	`mailto:mail@site.ru`
FTP	`ftp://ftp.site.ru/`
News	`news:news.users.information`
Gopher	`gopher://gopher.site.ru/`

Типы файлов для перехода по ссылке

Составляя гиперссылку, мы подразумеваем ссылку на HTML-документ (речь идет о службе WWW). Тем не менее, практика разработки электронных документов показывает актуальную необходимость в ссылках на файлы другого типа.

Например, на корпоративном Web-сайте можно создать ссылку на пример коммерческого договора, составленного в формате Microsoft Word (расширение doc), на сервере бесплатного программного обеспечения — на архив программы (расширения zip, rar и пр.), на ресурсе, содержащем коллекцию музыкальных файлов, - на аудиоролики в формате mp3 или RealPlayer.

Возникают два вопроса — какова структура ссылок для других типов файлов и каким образом такие документы будут открываться в браузере пользователя?

Что касается структуры организации гиперссылок, то она остается неизменной:

```
<A HREF="http://www.site.ru/docs/dogovor.doc">Коммерческий договор</A>
<A HREF="http://www.site.ru/download/program.zip">Архив программы</A>
<A HREF="http://www.site.ru/music/song.mp3">Музыкальный ролик</A>
```

В отношении загрузки запрашиваемого файла в формате, отличном от HTML, возможны три ситуации:

- браузер поддерживает запрашиваемый формат и откроет файл в своем рабочем окне (например, рисунки формата GIF, JPEG и пр.);
- браузер не поддерживает запрашиваемый формат и переадресовывает запрос операционной системе, которая запускает программу, предназначенную для просмотра файлов данного типа (например, видеоролики в формате MPEG4 вызовут программу Microsoft Windows Media);
- браузер не поддерживает запрашиваемый формат, который к тому же не соотносится ни с одной программой в настройках операционной системы. В этом случае браузер откроет стандартное диалоговое окно с возможными вариантами действий относительно открываемого файла неизвестного формата — **Open this file from its current location** (Открыть этот файл из текущего состояния) или Save **this file to disk** (Сохранить этот файл на диске). Аналогично браузер поступит и с некоторыми известными операционной системе форматами (например, с zip-архивами) (рис. 2.20).

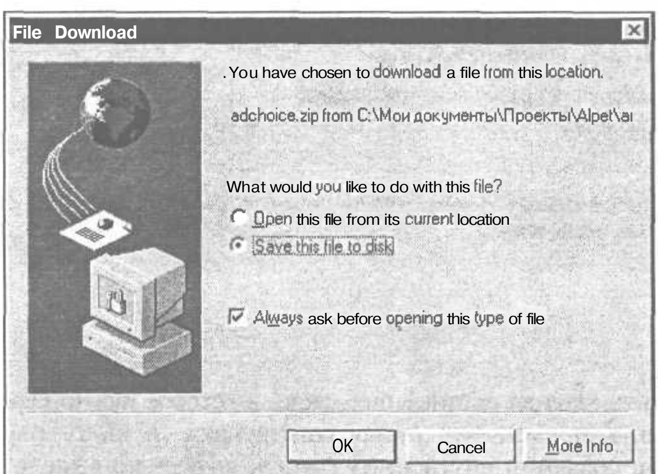


Рис. 2.20. Диалоговое окно браузера с вариантами действий относительно открываемого файла неизвестного формата

Внутренние ссылки

Если HTML-документ слишком большого размера и нет возможности разбить его на несколько отдельных файлов, можно прибегнуть к помощи внутренних гиперссылок, перемещающих пользователя в пределах одной Web-страницы. Такой подход бывает полезен, например, при составлении технической документации или юридических договоров, где быстрый переход от одного пункта документа к другому играет первостепенную роль.

Структура внутренней гиперссылки включает две части — сама ссылка и ее именной идентификатор (диез плюс имя элемента, аналогичное значению параметра HREF самой гиперссылки), позволяющий переместиться в нужное место электронного документа. Пример составления внутренних гиперссылок приведен в листинге 2.16.

Листинг 2,16. Пример составления внутренних гиперссылок

```
<HTML>
  <HEAD>
    <TITLE>Пример составления внутренних гиперссылок</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#FF0000"
VLINK="blue">
<FONT COLOR="gray" SIZE="6">Теги HTML</FONT><HR COLOR="#003366"
WIDTH="90%" ALIGN="left">
<PXA HREF="#font">FONT</A></P>
<PXA HREF="#span">SPAN</A></P>
<PXA HREF="#table">TABLE</A></P>
<HR ALIGN="center" WIDTH="95%" NOSHADE>
<P ALIGN="justify">
<A NAME="font"></A>
FONT — один из основных тегов физического форматирования текста, отобра-
жающий свойства шрифтов.
</P>
</BODY>
</HTML>
```

Следует заметить, что для обозначения места, в которое нужно перенести пользователя документа, применяется пустая конструкция <A NAME="font"></A>. Дело в том, что при создании внутренних ссылок визуально выделять расположение именного идентификатора не имеет смысла, потому что основная цель — переход в соответствующий раздел текущего документа.

Для лучшего усвоения материала основные параметры тега <A> приведены в табл. 2.6.

Таблица 2.6. *Параметры тега <A>*

Параметр	Функция
HREF	Указание адреса перехода по гиперссылке
TARGET	Определение места назначения перехода (текущее/новое окно)

Таблица 2.6 *(окончание)*

Параметр	Функция
STYLE	Задание стилевого шаблона
CLASS	Указание класса стилевого шаблона
NAME	Именной идентификатор внутренней гиперссылки

Итак, в этой главе мы рассмотрели основные параметры и свойства такого важнейшего компонента HTML-документа, как текст. Некоторые рассмотренные элементы собраны воедино в итоговом примере этой главы (листинг 2.17, рис. 2.21).

Листинг 2.17. Итоговый пример по главе 2

```
<HTML>
  <HEAD>
    <TITLE>Итоговый пример по главе 2</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#FF0000"
VLINK="blue">
<!-- Заголовок документа о программе Photoshop 7 -->
<H2>Adobe Photoshop 7</H2>
<DIVXA HREF="#intro">&raquo; Вступление</A></DIV>
<DIV><A HREF="#interface">&raquo; Интерфейс</A></DIV>
<DIVXA HREF="#system">&raquo; Системные требования</A></DIV>
<HR ALIGN="center" COLOR="gray">
<P ALIGN="justify"><FONT FACE="Tahoma" SIZE="3"><A NAME="intro"></A>
Начиная с пятой версии, разработчики корпорации
<A HREF="http://www.adobe.com/" TARGET="_blank">Adobe</A> старались
сделать свой программный продукт идущим в ногу со временем, другими
словами &mdash; ориентированным на стремительно развивающиеся
<NOBR>Web-технологии</NOBR>. В <STRONG>версии 7.0</STRONG> подобная
тенденция нисколько не ослабла, а скорее наоборот: претерпела новый бум
развития, что неизменно и <FONT COLOR="green"><BIG>положительным
образом</BIG></FONT> сказалось на самой программе.
<BRXBRXA NAME="interface"></A>
<A HREF="http://www.adobe.com/"><IMG SRC="photoshop.gif" BORDER="0"
WIDTH="64" HEIGHT="441" ALIGN="left">Прежде всего поменялся внешний вид
интерфейса</A> и сопутствующих элементов рабочей среды: подсвечивающиеся
кнопки инструментальной панели и значки форматирования объектов,
пиктограммы, перерисованные на новый лад (плавные края, чуть видные
градиентные заливки и пр.), новая программная заставка с использованием
```

```
прозрачности и др. <FONT FACE="Verdana" SIZE="2">Все это, безусловно,
радует глаз и делает визуальное восприятие рабочих возможностей пакета
более приятным.</FONT>

<BRXBRXA NAME="system"></A>

Несколько повысились <EM><B><FONT COLOR="#098EFA">системные требова-
ния</FONT></B></EM> к установке пакета Adobe Photoshop 7.0 на компьютер
пользователя: теперь необходимо иметь <CITE>Intel Pentium III или
4</CITE> (или другой процессор такого же класса), не менее 280 Мбайт сво-
бодного места на жестком диске, 128 Мбайт оперативной памяти (желательно
256 Мбайт), разрешение экрана не менее 800x600 точек с 16-битным цветом.

</FONTX/P>

</BODY>

</HTML>
```

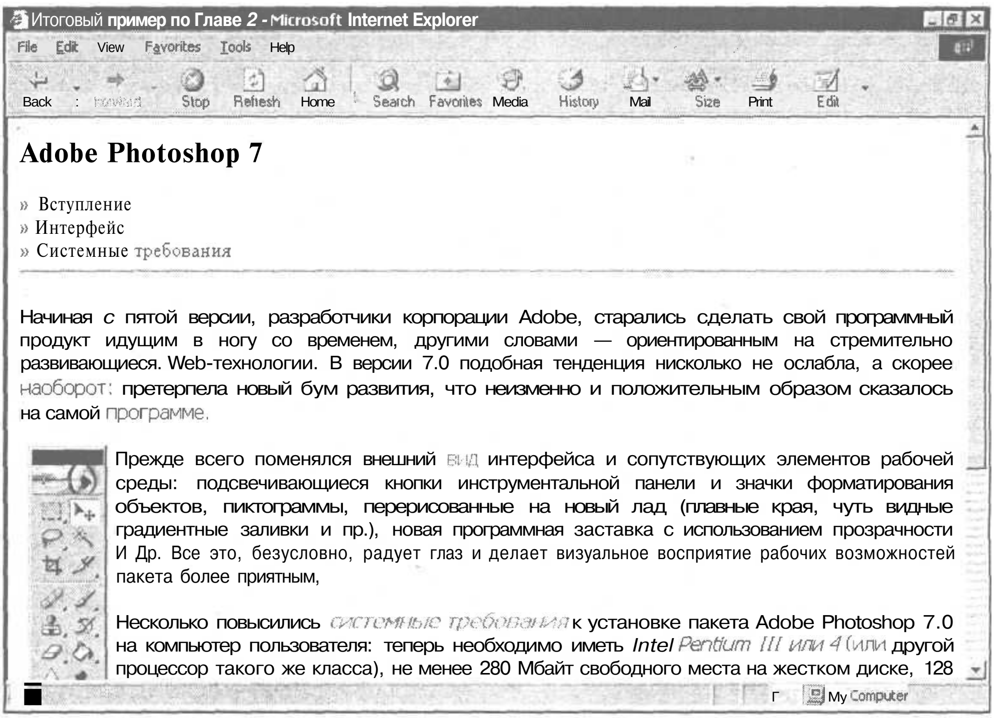


Рис. 2.21. Итоговый пример по главе 2

Глава 3

Структуризация информации

Зачем нужны HTML-списки?

Давайте задумаемся — как часто мы сталкиваемся в нашей повседневной жизни с перечнем какой-то информации? Грядет череда праздников и необходимо совершить множество всевозможных покупок для праздничного стола — мы составляем список всех необходимых продуктов. Планируя свой день, мы указываем в ежедневнике перечень дел, мероприятий и встреч. Очутившись в кафе, мы знакомимся с предложенным меню, состоящим также из списка наименований тех блюд и напитков, которые можем заказать.

Таким образом, различные списки играют далеко не последнюю роль в нашей повседневной жизни. Возникает вопрос — а почему мы так часто прибегаем к этому средству представления информации?

Во-первых. Информация в виде списка позволяет разбить большие массивы данных на отдельные, четкие фрагменты, которые человеку гораздо удобнее воспринимать, нежели весь поток целиком. Рассмотрим, например, визуальные эффекты графического редактора Adobe Photoshop 7.

- Drop Shadow. На объект накладывается тень, параметры которой (размер, цвет, расположение и насыщенность) можно изменять с помощью настроек эффекта.
- Bevel and Emboss. Эффект добавляет объекту объемность посредством наложения параметров выпуклости, вдавленности и затенения.
- Color Overlay. Заполнение объекта выбранным цветом с возможностью регулирования уровня насыщенности.

Во-вторых. Списки позволяют создавать понятную и логичную внутреннюю структуру информационных данных, ориентироваться в которой — дело простое и удобное. Например, в повестку дня входит обсуждение следующих вопросов:

- практическая необходимость покупки служебного автотранспорта;

- выделение дополнительных финансовых средств на составление рекламного бюджета;
- избрание состава Совета директоров компании.

В-третьих. Использование списков удобно для отображения определенных пошаговых и прочих последовательных процессов. Например, для начала работы с компьютером вам необходимо:

1. Поехать в близлежащий компьютерный магазин.
2. Приобрести персональный компьютер.
3. Привезти компьютер домой и соединить все его составляющие компоненты между собой.
4. Включить питание компьютера и нажать кнопку "Power".

При работе с текстом в различных программах типа Microsoft Word создание списков не представляется сложным делом — надо всего лишь указать область данных, которую текстовому редактору необходимо превратить в список.

В языке разметки HTML за организацию списков отвечает целый ряд тегов, конструкции которых должны соответствовать определенным правилам структуризации информационных данных с помощью HTML.

Спецификация HTML предусматривает три основных типа списков.

- Нумерованные списки
- Маркированные списки
- Списки определений

Рассмотрим каждый из названных типов более подробно.

Нумерованные списки

Нумерованный список часто называется упорядоченным (Ordered List) и представляет собой определенную последовательность элементов.

Для создания нумерованного списка используется тег-контейнер `<OL>`, внутри которого располагаются пункты перечня информационных данных, за указание которых отвечает специальный тег <LI> (List Item — пункт списка или элемент списка) (листинг 3.1, рис. 3.1).

Листинг 3.1. Пример нумерованного списка

```
<HTML>
  <HEAD>
    <TITLE>Пример нумерованного списка</TITLE>
```

```
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H3>Пример нумерованного списка</H3>
<HR ALIGN="center" NOSHADE WIDTH="98%">
<OL>
<FONT FACE="Tahoma"><B>Визуальные эффекты Adobe Photoshop 7:</B></FONT>
        <LI>Drop Shadow
        <LI>Bevel and Emboss
        <LI>Color Overlay
</OL>
</BODY>
</HTML>
```

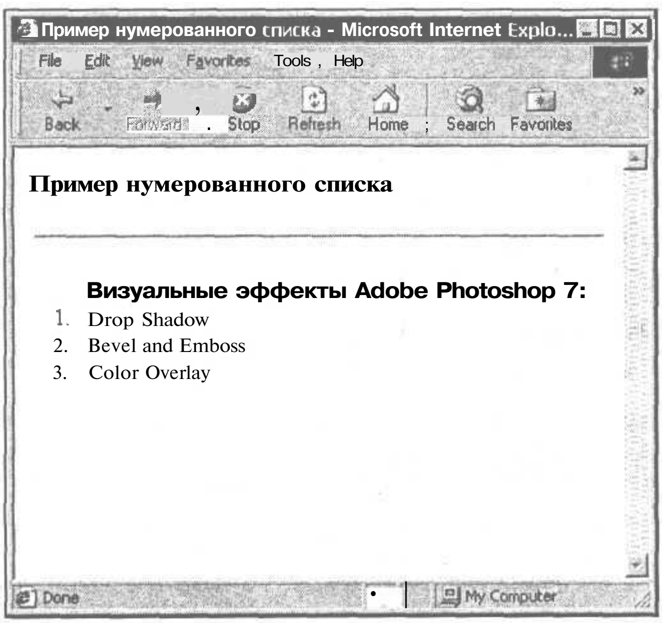


Рис. 3.1. Пример нумерованного списка

Как видно из примера, после каждого пункта списка осуществляется перенос строки, причем без указания каких-либо структурных тегов. Также следует заметить, что закрывающий тег </LI> отсутствует, хотя и существует в спецификации HTML (Консорциум W3C признал этот закрывающий тег необязательным).

Если заголовок списка "визуальные эффекты Adobe Photoshop 7:" размещается между тегом <OL> и первым пунктом списка <LI>, то он будет отделен

от начала самого списка одним переносом строки и отступом от левого края документа (как и все пункты списка). При указании заголовка перед тегом <OL> браузер, интерпретируя HTML-код, создаст двойной перенос и разместит текст заголовка левее пунктов списка (рис. 3.2).

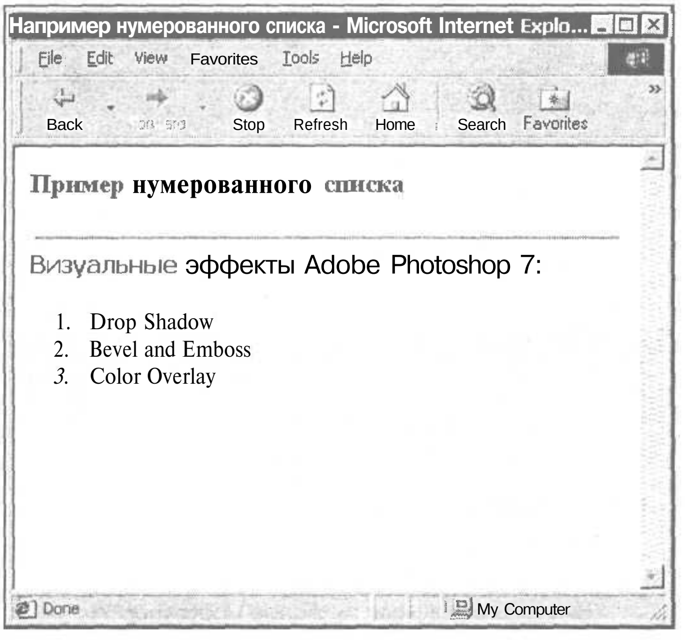


Рис. 3.2. Пример нумерованного списка с заголовком, размещенным перед тегом <OL>

Иногда может возникнуть ситуация, когда список содержит определенный разрыв в последовательности представления пунктов (например, после пункта 3 должен идти пункт 5). Для этой цели используется специальный параметр тега <LI> - VALUE (листинг 3.2, рис. 3.3).

! Листинг 3.2. Пример нумерованного списка с разрывом последовательности

```
<HTML>
  <HEAD>
    <TITLE>Пример нумерованного списка с разрывом последовательно-
сти</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H3>Пример нумерованного списка</H3>
```

```
<HR ALIGN="center" NOSHADE WIDTH="98%">
<FONT FACE="Tahoma"><B>Визуальные эффекты Adobe Photoshop 7:</B></FONT>
<OL>
        <LI>Drop Shadow
        <LI>Bevel and Emboss
        <LI>Color Overlay
        <BR>...
        <LI VALUE="5">Stroke
</OL>
</BODY>
</HTML>
```

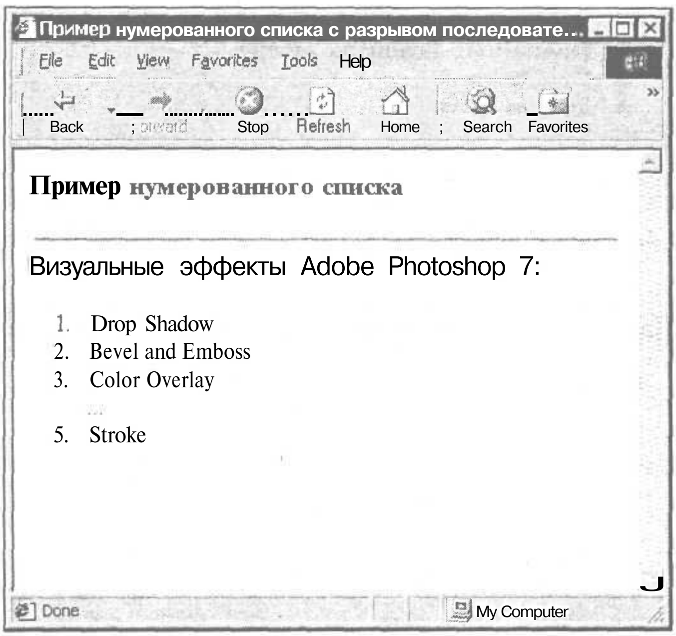


Рис. 3.3. Пример нумерованного списка с разрывом последовательности

Параметры тега <OL>

Тег составления нумерованных списков <OL> может включать параметры TYPE, START И COMPACT.

Параметр TYPE используется для указания типа нумерованного списка. Помимо обычных арабских цифр могут применяться большие и маленькие римские цифры, а также строчные и прописные латинские буквы (рис. 3.4, листинг 3.3).

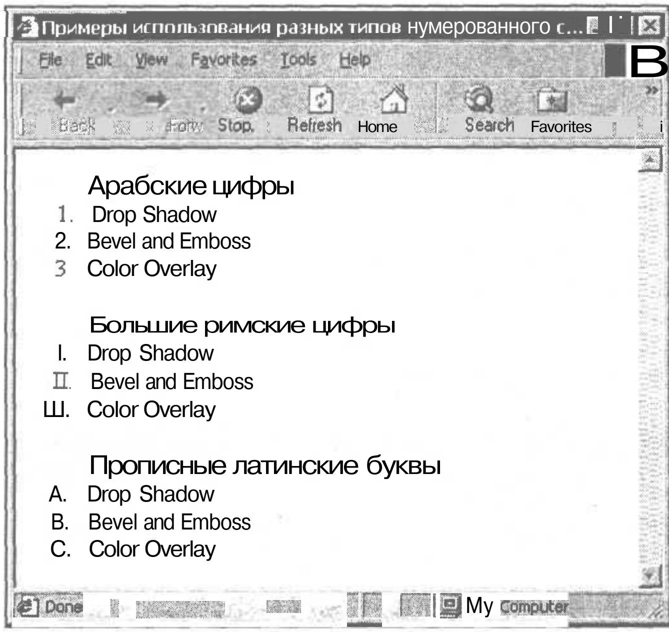


Рис. 3.4. Примеры использования разных типов нумерованного списка

Листинг 3.3. Примеры использования разных типов нумерованного списка

```
<HTML>
  <HEAD>
    <TITLE>Примеры использования разных типов нумерованного спи-
ска</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<OL TYPE="1">
<FONT SIZE="4"><B>Арабские цифры</B></FONT>
        <LI>Drop Shadow
        <LI>Bevel and Emboss
        <LI>Color Overlay
</OL>
<OL TYPE="I">
<FONT SIZE="4"><B>Большие римские цифры</B></FONT>
        <LI>Drop Shadow
        <LI>Bevel and Emboss
```

```
        <LI>Color Overlay
</OL>
<OL TYPE="A">
<FONT SIZE="4"><B>Прописные латинские буквы</B></FONT>
        <LI>Drop Shadow
        <LI>Bevel and Emboss
        <LI>Color Overlay
</OL>
</BODY>
</HTML>
```

Все значения параметра TYPE приведены в табл. 3.1.

Таблица 3.1. *Значения параметра TYPE тега <OL>*

Значение	Описание
TYPE="1"	Нумерация арабскими цифрами
TYPE="I"	Нумерация большими римскими цифрами
TYPE="i"	Нумерация маленькими римскими цифрами
TYPE="A"	Нумерация прописными латинскими буквами
TYPE="a"	Нумерация строчными латинскими буквами

Типом нумерации, используемым по умолчанию, является TYPE="1", поэтому при создании нумерованных списков такого типа необязательно задавать значение параметра TYPE (сам параметр также можно опустить).

Параметр START позволяет начинать нумерованный список не с первой позиции. При этом значением этого параметра может быть только числовой эквивалент конкретного типа списка. Например, чтобы начать список, созданный на основе нумерации большими латинскими буквами, с буквы F, надо указать следующую HTML-конструкцию:

```
<OL TYPE="A" START="6">
```

В данном случае значением параметра START будет являться цифра 6, т. к. буква F — шестая по счету буква в латинском алфавите.

Параметр COMPACT изначально был предусмотрен для списков, отображение которых должно производиться в компактном виде, например, с использованием меньшего размера шрифта, меньшего отступа от края страницы или меньшего расстояния между пунктами списка. Однако на практике этот параметр почти не используется, поскольку наиболее распространенные браузеры никак не реагируют на его применение, к тому же Консорциум W3C в последней спецификации HTML вообще не рекомендует использовать параметр COMPACT.

В завершение разговора о создании нумерованных списков следует добавить, что возможно смешанное использование различных типов нумерации в пределах одного списка. Например, первый пункт списка можно задать типом арабских цифр, второй — буквенным способом, третий — с помощью римских цифр и т. д.

Однако браузеры по-разному реагируют на такое смешение стилей отображения списков. К примеру, HTML-конструкция:

```
<OL TYPE="1">
        <LI>Drop Shadow
        <LI TYPE="A">Bevel and Emboss
        <LI TYPE="i">Color Overlay
        <LI>Outer Glow
        <LI TYPE="I">Stroke
</OL>
```

будет интерпретирована браузером Internet Explorer так, как показано на рис. 3.5 — в списке четвертый по счету пункт "Outer Glow" озаглавлен арабской цифрой 4.

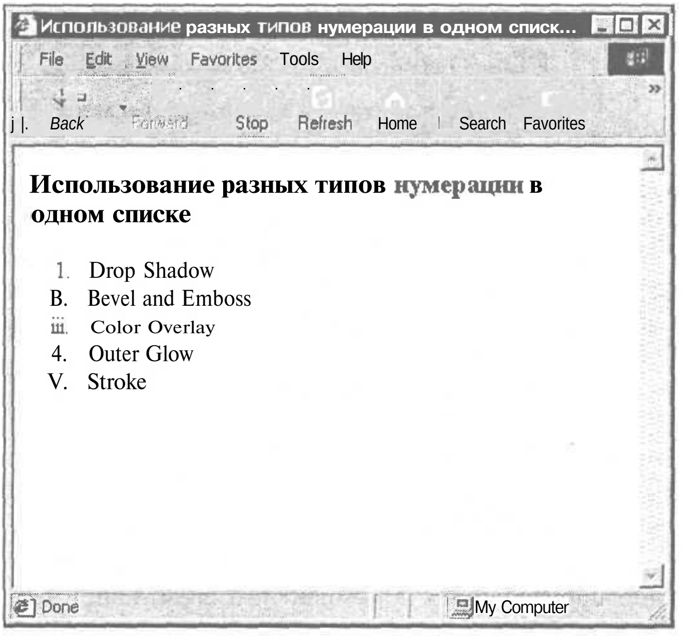


Рис. 3.5. Отображение списка со смешанной нумерацией в Internet Explorer

А вот другой браузер — Netscape Navigator — расценит пропуск типа нумерации в пункте "Outer Glow" как команду отображать данные по предыдущему типу (маленькие римские цифры) (рис. 3.6).

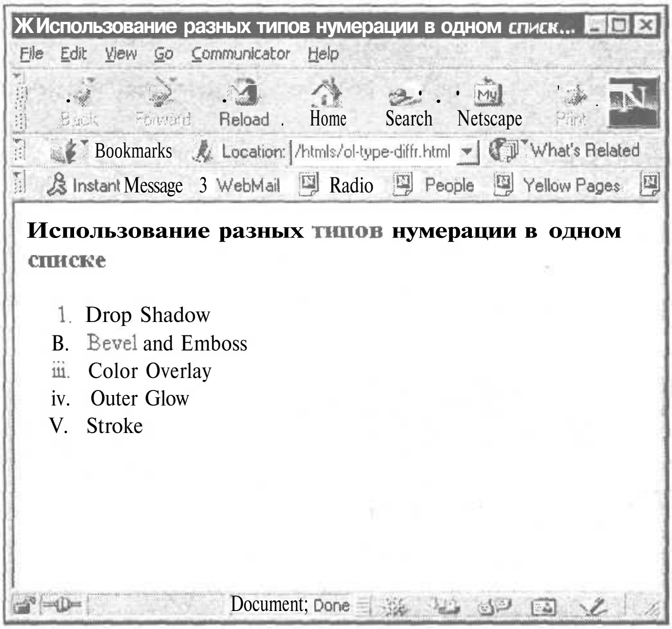


Рис. 3.6. Отображение списка со смешанной нумерацией в Netscape Navigator

Маркированные списки

Другим распространенным способом структуризации информационных данных является создание маркированных списков. Маркированный список представляет собой ненумерованный или неупорядоченный (Unordered List) перечень элементов, для заголовка которых используются специальные маркеры. В качестве маркеров выступают специальные символы, называемые буллетами (bullets).

Для создания маркированных списков применяется тег-контейнер `<UL>`, внутри которого располагаются элементы самого списка (как и в случае с нумерованными списками, используется тег `<LI>`, не требующий закрывающего тега) (листинг 3.4, рис. 3.7).

Листинг 3.4. Пример маркированного списка

```
<HTML>
  <HEAD>
    <TITLE>Пример маркированного списка</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
```

```
<H3>Пример маркированного списка</H3>
<HR ALIGN="center" NOSHADE WIDTH="98%">
<FONT FACE="Tahoma"><B>Популярные редакторы векторной графики:</B></FONT>
<UL>
	<LI>CorelDRAW
	<LI>Adobe Illustrator
	<LI>Macromedia FreeHand
</UL>
</BODY>
</HTML>
```

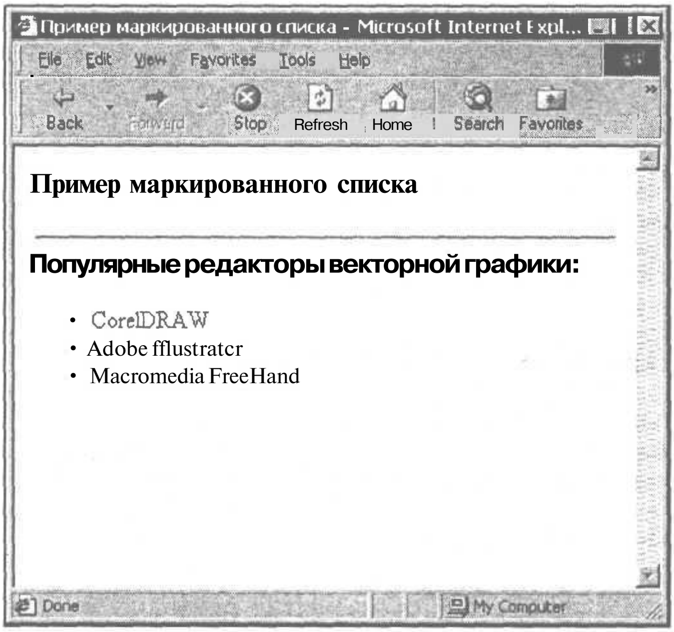


Рис. 3.7. Пример маркированного списка

Параметры тега *<UL>*

Тег маркированного списка <UL> может содержать два основных параметра: TYPE и COMPACT. Параметр TYPE отвечает за форму отображаемого браузером маркера. Всего существует три возможных значения данного параметра:

- TYPE=disc — значение, используемое по умолчанию (указание необязательно), которое отображает маркер в виде закрашенного круга;
- TYPE=circle — маркер отображается в виде незакрашенного круга (кольца);
- TYPE=square — отображение маркера в виде закрашенного квадрата.

Примеры различных типов маркированного списка приведены соответственно на рис. 3.8 и в листинге 3.5.

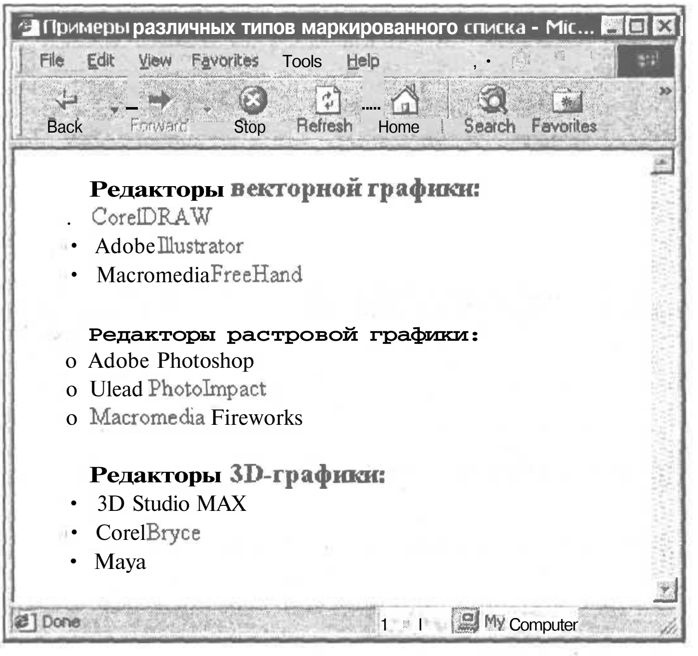


Рис. 3.8. Примеры различных типов маркированного списка

Листинг 3.5. Примеры различных типов маркированного списка

```
<HTML>
  <HEAD>
    <TITLE>Примеры различных типов маркированного списка</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<UL>
<FONT SIZE="4"><B>Редакторы векторной графики:</B></FONT>
       <LI>CorelDRAW
       <LI>Adobe Illustrator
       <LI>Macromedia FreeHand
</UL>
<UL TYPE="circle">
<FONT SIZE="4"><B>Редакторы растровой графики:</B></FONT>
```

```
        <LI>Adobe Photoshop
        <LI>Ulead PhotoImpact
        <LI>Macromedia Fireworks
</UL>
<UL TYPE="square">
<FONT SIZE="4"><B>Редакторы 3D-графики:</BX/FONT>
        <LI>3D Studio MAX
        <LI>Corel Bryce
        <LI>Maya
</UL>
</BODY>
</HTML>
```

Следует отметить, что размер того или иного маркера может изменяться в зависимости от размера используемого в электронном документе шрифта. Кроме этого, как и в случае с нумерованными списками, при смешении правил маркировки различные браузеры могут отображать маркеры по-разному.

Параметр тега `<UL>` COMPACT имеет те же характеристики, что и для нумерованных списков.

Графические маркеры

Отличительной особенностью маркированных списков является возможность использования вместо стандартных HTML-маркеров свои собственные графические изображения. Такой подход позволяет создавать более привлекательные списки в электронных документах.

Для указания браузеру того, что в качестве маркера будет использовано графическое изображение, внутри тега-контейнера `<UL>` вместо указателя элемента `<LI>` размещается обыкновенная HTML-конструкция описания графических объектов (листинг 3.6, рис. 3.9).

Листинг 3.6. Пример использования графических маркеров

```
<HTML>
  <HEAD>
    <TITLE>Пример использования графических маркеров</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<UL>
```

```
<FONT SIZE="4"><B>Редакторы компьютерной графики:</B></FONT>
        <IMG SRC="marker.gif">CorelDRAW
        <BR>
        <IMG SRC="marker.gif">Adobe Illustrator
        <BR>
        <IMG SRC="marker.gif">Macromedia FreeHand
        <BR>
        <IMG SRC="marker.gif">Adobe Photoshop
        <BR>
        <IMG SRC="marker.gif">Macromedia Fireworks
        <BR>
        <IMG SRC="marker.gif">3D Studio MAX
        <BR>
        <IMG SRC="marker.gif">Corel Bryce
</UL>
</BODY>
</HTML>
```

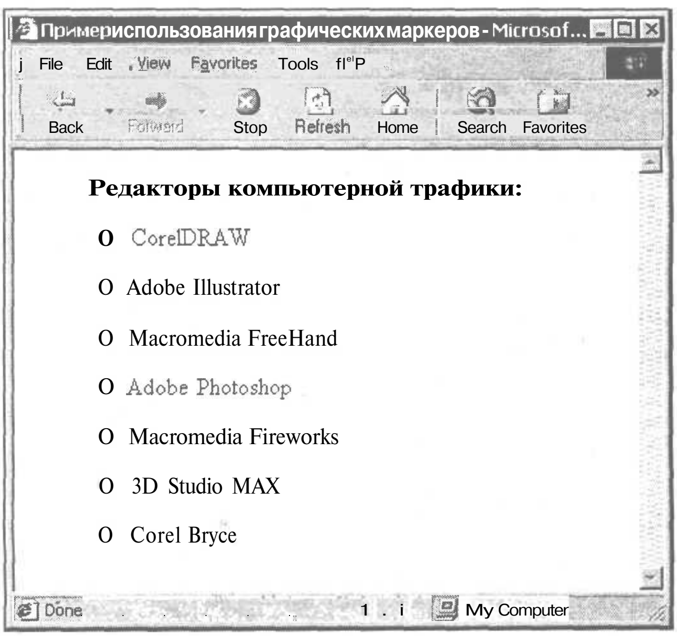


Рис. 3.9. Пример использования графических маркеров

Браузер поймет такую конструкцию как команду задать отступ от левого края документа до начала элементов маркированного списка (где повторяющимся маркером является файл рисунка "marker.gif").

В данном случае после каждого пункта маркированного списка стоит тег переноса строки
. Если его не указывать, все элементы списка будут выстроены в один ряд. Это связано с тем, что создание графических маркеров не относится к каким-то особенным HTML-конструкциям, а реализовано за счет обособления рисунков-маркеров тегом <UL>. Поскольку тег элемента списка <LI> отсутствует, браузер не делает переноса строки.

Вложенные списки

Часто информация имеет сложную разветвленную структуру, которую с одной стороны трудно отобразить в виде одного списка, с другой — нельзя разбивать на несколько отдельных списков. В этом случае используются так называемые вложенные списки, имеющие иерархическое строение и содержащие в себе другие списки (как нумерованные, так и маркированные) (листинг 3.7, рис. 3.10).

Листинг 3.7. Пример использования вложенных списков

```
<HTML>
  <HEAD>
    <TITLE>Пример использования вложенных списков</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<UL TYPE="square">
<FONT SIZE="4"><B>Редакторы компьютерной графики:</B></FONT>
        <LI>Векторные редакторы
                <UL TYPE="disc">
                <LI>CorelDRAW
                        <OL>
                        <LI>CorelDRAW
                        <LI>Corel PhotoPaint
                        <LI>Corel R.A.V.E.
                        </OL>
                </UL>
        <LI>Растровые редакторы
                <UL TYPE="disc">
                <LI>Adobe Photoshop
                        <OL>
                        <LI>Adobe Photoshop
                        <LI>Adobe ImageReady
```

```
                        </OL>
                </UL>
</UL>
</BODY>
</HTML>
```

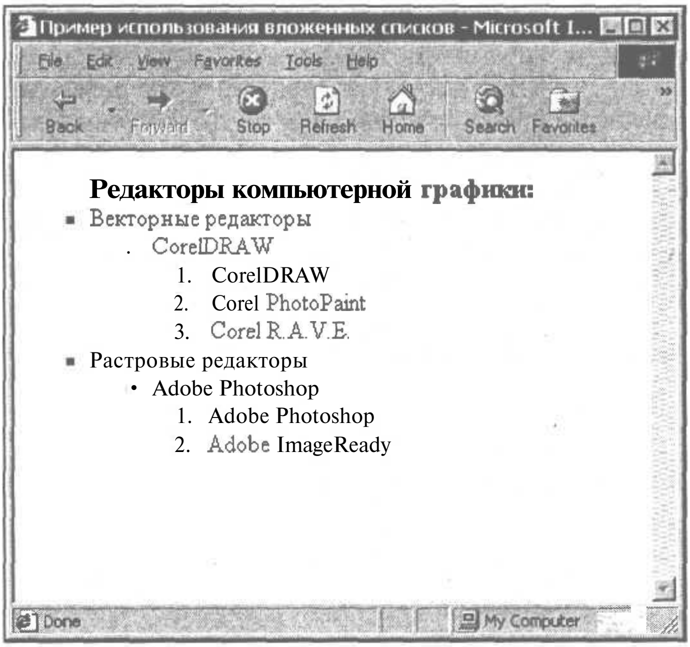


Рис. 3.10. Пример использования вложенных списков

В данном примере присутствуют и нумерованные, и маркированные списки. Графические маркеры также могут применяться внутри вложенных списков наравне со списками остальных типов.

В заключение необходимо добавить, что при составлении сложных вложенных списков особое внимание следует уделять корректному указанию закрывающих тегов `</UL>` и `</OL>`. Для того чтобы не запутаться в разветвленной структуре HTML-кода вложенных списков, каждый последующий уровень вложенности лучше всего отделять от предыдущего небольшим отступом от левого края документа.

Списки определений

Списки определений — это особый тип структуризации информационных данных, идеально подходящий для отображения терминологических и толковых словарей, а также различных справочников средствами HTML.

В отличие от нумерованных и маркированных списков, которые могут содержать помимо самих элементов еще и другие списки, список определений состоит всего из двух частей: самого определения (термина) и его содержательной части (описания).

Структура списка определений описывается тегом-контейнером <DL> (Definition List), внутри которого указывается тег <DT> (Definition Term), выделяющий заголовок определения, и тег <DD> (Definition Description), описывающий содержательную часть определения. Для двух последних тегов наличие закрывающих тегов необязательно. Пример использования списков определений приведен соответственно на рис. 3.11 и в листинге 3.8.

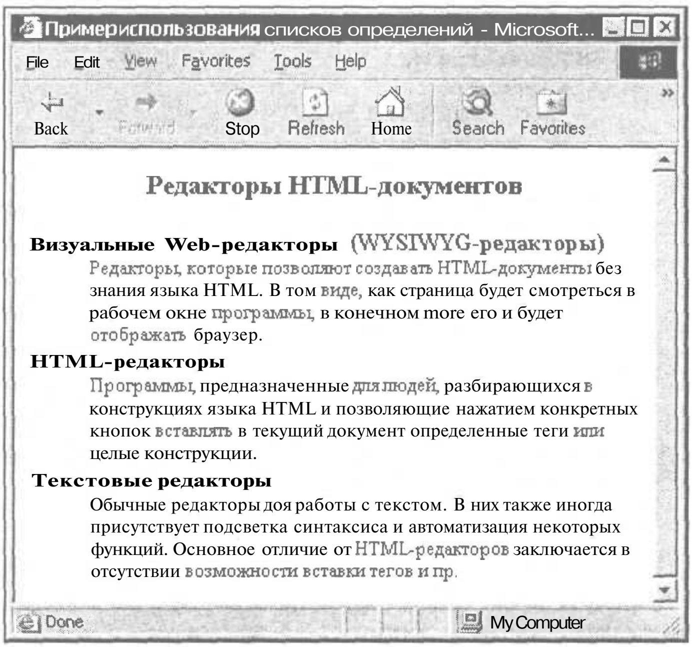


Рис. 3.11. Пример использования списков определений

Листинг 3.8. Пример использования списков определений

```
<HTML>
  <HEAD>
    <TITLE>Пример использования списков определений</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<DL>
```

```
<H3 ALIGN="center">Редакторы HTML-документов</H3>
<DT><B>Визуальные Web-редакторы (WYSIWYG-редакторы)</B>
<DD><SMALL>Редакторы, которые позволяют создавать HTML-документы без знания языка HTML. В том виде, как страница будет смотреться в рабочем окне программы, в конечном итоге его и будет отображать браузер.</SMALL><BR>
<DT><B>HTML-редакторы</B>
<DD><SMALL>Программы, предназначенные для людей, разбирающихся в конструкциях языка HTML и позволяющие нажатием конкретных кнопок вставлять в текущий документ определенные теги или целые конструкции.</SMALL><BR>
<DT><B>Текстовые редакторы</B>
<DD><SMALL>Обычные редакторы для работы с текстом. В них также иногда присутствует подсветка синтаксиса и автоматизация некоторых функций. Основное отличие от HTML-редакторов заключается в отсутствии возможности вставки тегов и пр.</SMALL><BR>
</DL>
</BODY>
</HTML>
```

Как видно из листинга 3.8, заголовок определения и содержательная часть списка определений позволяют применять форматирование текста, а также некоторые теги структурного форматирования. Также видно, что каждое определение в коде разделено тегом переноса строки
. Если этого не сделать, большинство браузеров создаст между последней и предпоследней строкой содержательной части определения промежуток, больший по размеру, чем между остальными строками описания.

Помимо этого, в структуру списков определений можно включать другие типы списков — нумерованные и маркированные (в т. ч. с графическими маркерами).

Тег структуры списка определений <DL> может содержать параметр COMPACT, суть которого была описана в разделах этой главы, посвященных нумерованным и маркированным спискам.

Другие списки

К спискам других типов можно отнести списки, описываемые тегами <DIR> и <MENU>. Изначально такие списки планировалось делать более компактными и строгими по отношению к какому-либо форматированию. Также предполагалось наложить ряд ограничений: на длину строки элементов списка, на количество символов в тексте и пр.

Тем не менее, в последней спецификации HTML списки этих типов были названы отмененными, несмотря на их поддержку большинством современ-

ных браузеров. Вместо <DIR> и <MENU> Консорциум W3C рекомендует использовать стандартные маркированные списки.

Для закрепления полученного в этой главе материала все описанные типы HTML-списков приведены в табл. 3.2.

Таблица 3.2. *Типы HTML-списков*

Тип списка	**Описание**	**Используемые теги/параметры**
Нумерованный список	Упорядоченный список с возможностью нумерации по буквам и цифрам	<OL>, <LI> TYPE, START, COMPACT
Маркированный список	Неупорядоченный список с возможностью маркировки элементов (в т. ч. графическими изображениями)	<UL>, <LI> TYPE, COMPACT
Список определений	Список терминов с содержательной частью (описанием термина)	<DL>, <DT>, <DD> COMPACT

Глава 4

Таблица – основа вашего HTML-документа

Табличное представление данных

Одним из наиболее мощных и гибких средств представления информационных данных в HTML по праву являются таблицы. В повседневной жизни, решая те или иные задачи, мы часто сталкиваемся с таблицами. Однако в HTML таблицы не ограничиваются удобным средством структурирования информации. Сегодня таблица становится основой большинства электронных документов, структура которых может включать самые разнообразные элементы HTML. Таблицы, изначально избранные в качестве визуального способа представления данных, сейчас имеют гораздо более важную функцию — управление структурой HTML-документа в целом. Удобство размещения данных в таблице неоспоримо, а преимущества перед другими средствами представления информации (например, списки) позволяют считать таблицы основополагающим структурным элементом любого HTML-документа.

Однако первостепенной задачей таблицы все же является представление информационных данных. Поэтому далее речь пойдет о том, как правильно создавать таблицы (подробнее о таблицах, используемых в качестве структурной основы HTML-документов, будет рассказано в *разд. "Вложенные таблицы" этой главы*).

Создание простейших таблиц

HTML-конструкция таблицы может содержать множество различных тегов и параметров, часть которых распространена, а часть используется лишь в редких случаях.

Основным тегом-контейнером таблицы является тег <TABLE>, который требует обязательного присутствия закрывающего тега. Любая таблица состоит из ряда (тег-контейнер <TR>, Table Row), содержащего определенное количество ячеек (тег-контейнер <TD>, Table Data). В принципе тег `<TD>` предназначен

для указания данных в таблице, а для размещения заголовков в ячейке используется тег <TH> (Table Header). Различие между этими двумя тегами заключается в визуальном представлении данных внутри таблицы: по умолчанию текст, размещенный в теге <TH>, отображается жирным начертанием с типом выравнивания по центру, а текст тега <TD> — обычным начертанием с выравниванием по левому краю.

Теги <TR>, <TD> и <TH> могут не иметь своих закрывающих тегов. Однако все же рекомендуется их указывать для предотвращения ошибок, которые могут возникнуть при создании сложных вложенных таблиц. Тег <TABLE>, как уже было сказано, обязательно должен иметь закрывающий тег.

Таблица может состоять из любого количества рядов (<TR>), каждый из которых может включать любое количество ячеек (<TD>, <TH>). При этом один ряд таблицы не может содержать ячейки другого ряда.

Таблице можно присвоить заголовок с помощью тега-контейнера <CAPTION>, который указывается сразу же после тега <TABLE> (в принципе, расположение тега <CAPTION> возможно в любом месте основного тега таблицы, за исключением областей описания тегов рядов и ячеек, однако в спецификации такой подход не приветствуется). Пример простейшей таблицы приведен соответственно в листинге 4.1 и на рис. 4.1.

Листинг 4.1. Пример простейшей таблицы

```
<HTML>
  <HEAD>
    <TITLE>Пример простейшей таблицы</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<TABLE BORDER>
<CAPTION>Пример простейшей таблицы</CAPTION>
<TR>
<TH>Ячейка с заголовком</TH>
<TD>Ячейка с обычным текстом</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

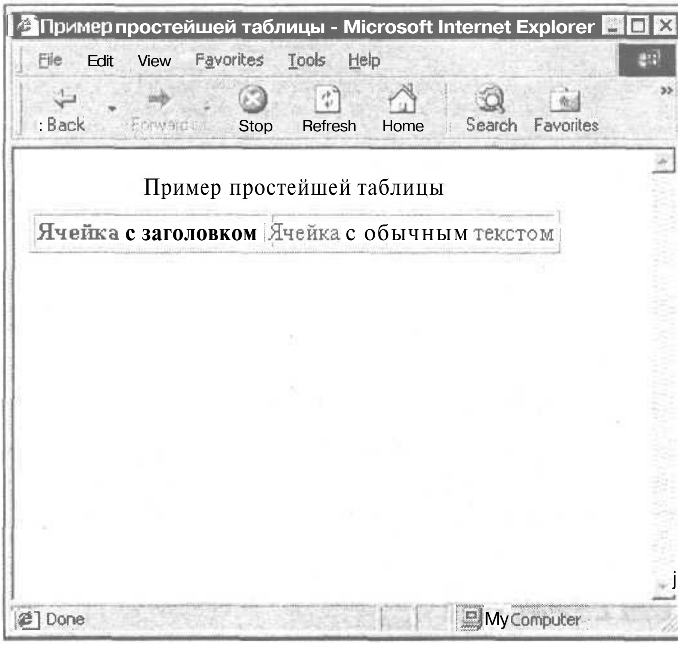


Рис. 4.1. Пример простейшей таблицы

Правила описания таблиц

В этом разделе более подробно будет рассказано об основных параметрах (а также их возможных значениях) упомянутых тегов таблицы.

Параметры тега <*TABLE*>

Тег <TABLE> является основным тегом таблицы и может содержать ряд параметров, каждый из которых не обязателен для указания. Следует заметить, что некоторые популярные браузеры (Internet Explorer, Netscape и др.) поддерживают часть дополнительных параметров таблицы. Далее речь пойдет о наиболее распространенных параметрах тега <TABLE>, поддерживаемых большинством современных браузеров.

Параметр *BORDER*

С помощью параметра BORDER можно задавать рамку, показывающую внутреннюю структуру таблицы, которая состоит из определенного количества рядов и ячеек. Отсутствие данного параметра означает, что рамка не будет отображаться браузером. Необходимость в таком варианте может возникнуть, например, при составлении больших списков или при создании таблицы в качестве основы HTML-документа.

Значением параметра BORDER может быть любое целое число, указываемое в пикселах:

```
<TABLE BORDER="2">
<TR>
<TD>Ячейка 1</TD>
<TD>Ячейка 2</TD>
</TR>
</TABLE>
```

При отсутствии установленного значения параметр BORDER принимает вид BORDER="1".

Параметр *BORDERCOLOR*

Параметр BORDERCOLOR предназначен для определения цвета рамки вокруг таблицы (поддерживается только Internet Explorer). Значение указывается в шестнадцатеричном формате (например, #FFCC00) или в виде наименования (например, green).

Параметр *CELLSPACING*

Параметр CELLSPACING задает расстояние между соседними ячейками таблицы по вертикали и горизонтали. Значение параметра указывается в пикселах и не может быть пропущено. При конструкции CELLSPACING="0" ячейки таблицы сольются, т. к. параметр устанавливает нулевое значение расстояния между соседними ячейками.

```
<TABLE BORDER="2" CELLSPACING="2">
<TR>
<TD>Ячейка 1</TD>
<TD>Ячейка 2</TD>
</TR>
</TABLE>
```

Расстояние между соседними ячейками (как по вертикали, так и по горизонтали) всегда будет одинаково, т. к. HTML не позволяет назначать разные значения в пределах данного параметра.

Параметр *CELLPADDING*

Для создания и регулирования отступа между рамкой ячейки и ее содержимым (по вертикали и горизонтали) используется параметр CELLPADDING. Форма указания значения аналогична параметру CELLSPACING и также не может быть пропущена.

```
<TABLE BORDER="2" CELLSPACING="2" CELLPADDING="7">
<TR>
```

```
<TD>Ячейка 1</TD>
<TD>Ячейка 2</TD>
</TR>
</TABLE>
```

При значении параметра CELLPADDING, равном нулю, содержимое ячейки будет вплотную прижато к рамке таблицы, что ухудшит восприятие информации (это утверждение еще более актуально для таблиц с прозрачной рамкой, т. е. значением BORDER="0", — в этом случае текст соседних ячеек сольется друг с другом).

Отступ между рамкой ячейки и ее содержимым (как по вертикали, так и по горизонтали) всегда будет одинакового размера (как и в случае с параметром CELLSPACING), т. к. HTML не позволяет назначать разные значения в пределах данного параметра.

Параметр *ALIGN*

Данный параметр устанавливает тип выравнивания всей таблицы относительно ширины HTML-документа. Существует три возможных значения параметра ALIGN:

- ALIGN="left" -- выравнивание таблицы по левому краю документа. Значение принято по умолчанию и в случае отсутствия параметра ALIGN таблице будет присвоено именно это значение;
- ALIGN="right" -- выравнивание таблицы по правой границе документа;
- ALIGN="center" — центрирование таблицы относительно ширины документа.

В некоторых учебниках и руководствах по языку разметки HTML утверждается, что выравнивание таблицы по центру не поддерживается наиболее популярными браузерами. На самом деле это неверно — все три наиболее распространенных браузера (Internet Explorer, Netscape и Opera), описание которых можно найти в *разд. "Ваш первый HTML-документ" главы 1,* поддерживают центрирование таблицы по горизонтали. Пример выравнивания таблицы по центру приведен в листинге 4.2 (рис. 4.2 демонстрирует выравнивание браузером Internet Explorer, рис. 4.3 браузером Netscape, рис. 4.4 — браузером Opera).

Листинг 4.2. Пример выравнивания таблицы по центру

```
<HTML>
  <HEAD>
    <TITLE>Пример выравнивания таблицы по центру</TITLE>
  </HEAD>
```

```
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue" LEFTMARGIN="40" RIGHTMARGIN="40" MARGINWIDTH="40">
<TABLE ALIGN="center" BORDER="2" CELLSPACING="2" CELLPADDING="7">
<TR>
<TH>Состав персонального компьютера</TH>
<TD>
<FONT FACE="Tahoma">В состав персонального компьютера входят следующие
компоненты:</FONT>
<UL TYPE="disc">
        <LI>Системный блок
        <LI>Монитор
        <LI>Клавиатура
        <LI>Мышь
</UL>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

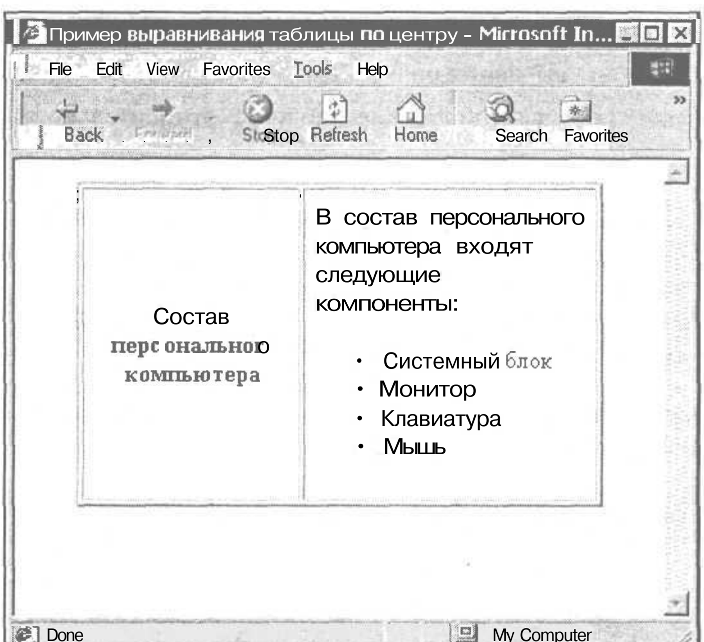


Рис. 4.2. Пример выравнивания таблицы по центру (Internet Explorer)

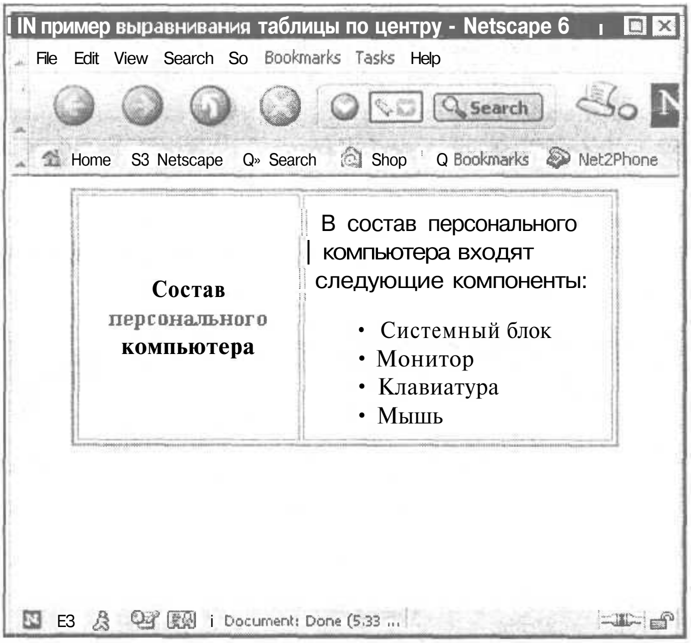


Рис. 4.3. Пример выравнивания таблицы по центру (Netscape)

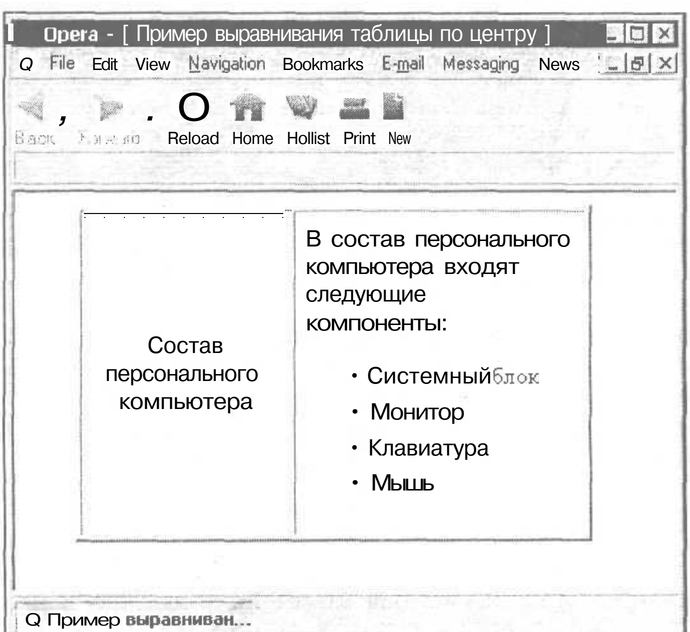


Рис. 4.4. Пример выравнивания таблицы по центру (Opera)

Как видно из рисунков 4.2—4.4, отступы между таблицей и фаницами документа во всех трех случаях равны, что подтверждает справедливость высказывания относительно поддержки выравнивания таблицы по центру наиболее распространенными браузерами (при сравнении отступов не следует принимать во внимание отступ, оставленный Internet Explorer для возможной полосы прокрутки).

Помимо функции выравнивания таблицы по горизонтали документа, параметр ALIGN позволяет получать эффект "обтекания" таблицы текстом (рис. 4.5).

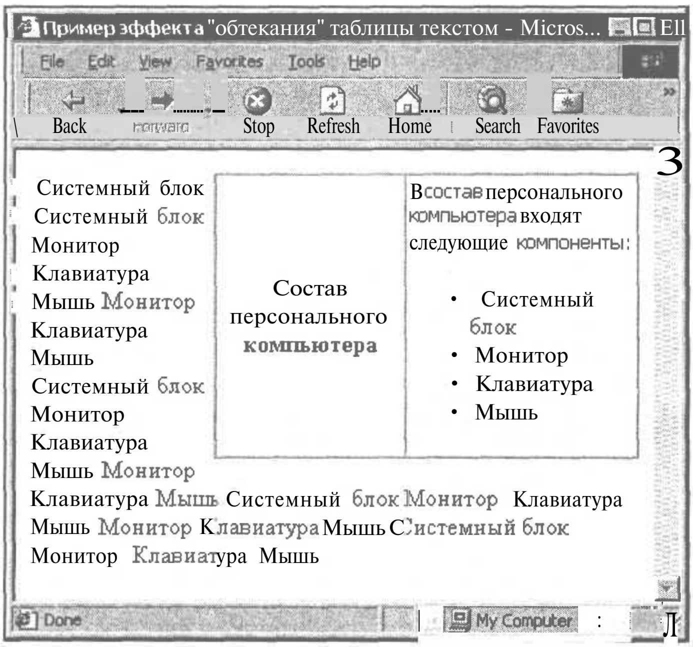


Рис. 4.5. Пример эффекта "обтекания" таблицы текстом

Для получения такого эффекта нужно выбрать тип выравнивания таблицы (в данном случае, кроме центрирования), назначить ширину таблицы менее 100% или зафиксировать ее (подробнее об изменении параметра ширины таблицы будет рассказано чуть позже). Текст, который должен огибать таблицу, размещается сразу же после закрывающего тега </TABLE>.

Параметры *WIDTH и HEIGHT*

Параметры WIDTH и HEIGHT отвечают соответственно за ширину и высоту таблицы. Значение этих параметров может быть указано в процентах или пикселах:

```
<TABLE WIDTH="500" HEIGHT="50%">
```

В случае указания ширины или высоты в пикселах таблица будет иметь фиксированные размеры. Такая таблица при изменении размеров окна браузера останется неизменной. При использовании в качестве единицы измерения процентов ширина/высота таблицы будет варьироваться в зависимости от размеров окна браузера. В этом случае размер ячеек таблицы будет уменьшаться/увеличиваться пропорционально заявленному значению в процентах. Содержимое таблицы будет перемещаться в окне браузера соответственно изменению его размеров. Например, при уменьшении окна браузера текст, размещенный в ячейке, может быть перенесен на другую строку, растянут или, наоборот, сжат (при выравнивании текста по ширине: P ALIGN="justify").

Параметры тега <TABLE> перечислены в табл. 4.1.

Таблица 4.1. *Параметры тега <TABLE>*

Параметр	Функция
BORDER	Создание рамки вокруг таблицы
BORDERCOLOR	Определение цвета рамки вокруг таблицы
CELLSPACING	Указание расстояния между соседними ячейками
CELLPADDING	Создание отступа от границы ячейки до ее содержимого-
ALIGN	Указание типа выравнивания таблицы по ширине окна браузера
WIDTH	Определение ширины таблицы
HEIGHT	Определение высоты таблицы

Параметры тегов *<TR>*, *<TD>* и *<TH>*

Теги <TR>, <TD> и <TH> составляют внутреннюю структуру таблицы и могут содержать набор определенных параметров. Прежде чем перейти к рассмотрению этих параметров, следует сказать, что внутри тега указания ряда таблицы <TR> не может располагаться текст, списки, графические изображения и прочие HTML-элементы и теги. Форматирование любых типов и указание других тегов может быть только в пределах тегов <TD> и <TH>, определяющих содержимое табличных ячеек.

Параметры *ALIGN* и *VALIGN*

Параметр ALIGN предназначен для указания типа выравнивания содержимого ячеек. Если в случае использования этого параметра в теге <TABLE> вся таблица выравнивается определенным образом, то в данном случае назначается тип выравнивания для конкретной ячейки или ряда таблицы. Если необходимо, чтобы каждая ячейка содержала разные типы выравнивания,

придется указывать соответствующее значение параметра ALIGN в каждом теге <TD> или <тн>. Если требуется задать один тип выравнивания для всего ряда (включающего все ячейки), значение параметра ALIGN прописывается в теге <TR>.

Возможными значениями параметра ALIGN являются ALIGN="left", ALIGN="right" и ALIGN="center". При этом для <TD> значением по умолчанию считается выравнивание по левому краю ячейки, для <тн> — выравнивание по центру.

Параметр VALIGN также определяет тип выравнивания содержимого ячеек таблицы, но по вертикали. Он может принимать следующие значения:

- VALIGN="middle" — выравнивание по середине ячейки (значение по умолчанию);
- VALIGN="top" — выравнивание по верхнему краю ячейки;
- VALIGN="bottom" — выравнивание по нижней границе ячейки;
- VALIGN="baseline" — выравнивание по базовой линии (привязка текста одной строки к единому уровню).

Пример разных типов выравнивания ячеек таблицы приведен соответственно в листинге 4.3 и на рис. 4.6.

Листинг 4.3. Разные типы выравнивания ячеек таблицы

```
<HTML>
  <HEAD>
    <TITLE>Разные типы выравнивания ячеек таблицы</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue" LEFTMARGIN="40" RIGHTMARGIN="40" MARGINWIDTH="40">
<TABLE ALIGN="center" BORDER="2" CELLSPACING="0" CELLPADDING="5"
WIDTH="100%" HEIGHT="200">
<TR ALIGN="center">
<TD VALIGN="bottom">Ячейка 1</TD>
<TD>Ячейка 2</TD>
<TD VALIGN="top">Ячейка 3</TD>
</TR>
<TR>
<TD align="right">Ячейка 4</TD>
<TD align="center">Ячейка 5</TD>
```

```
<TD>Ячейка 6</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

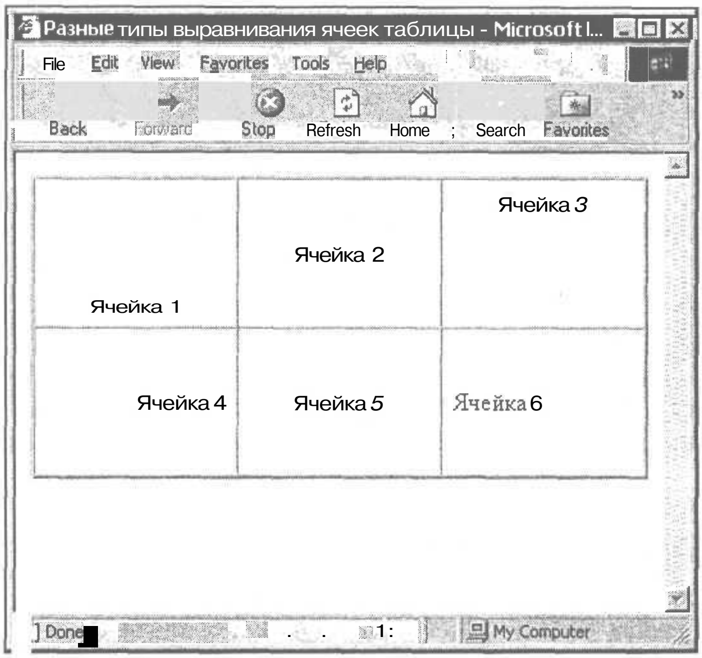


Рис. 4.6. Разные типы выравнивания ячеек таблицы

Параметры ALIGN и VALIGN используются в тегах <TR> (для всего ряда таблицы), <TD> и <тн>.

Параметры *BGCOLOR* и *BACKGROUND*

Параметры BGCOLOR и BACKGROUND используются для работы с фоном ячеек или целых рядов таблицы. BGCOLOR позволяет задать определенный цвет ячеек, а BACKGROUND — указать путь к графическому изображению, которое будет служить фоном (рис. 4.7).

Действие данных параметров в отношении тега ряда <TR> и тегов ячейки <TD> и <тн> аналогично типу выравнивания ALIGN и VALIGN.

Параметр BGCOLOR используется для тегов <TR>, <TD> и <тн>. Параметр BACKGROUND применим только к тегам ячеек <TD> и <TH>.

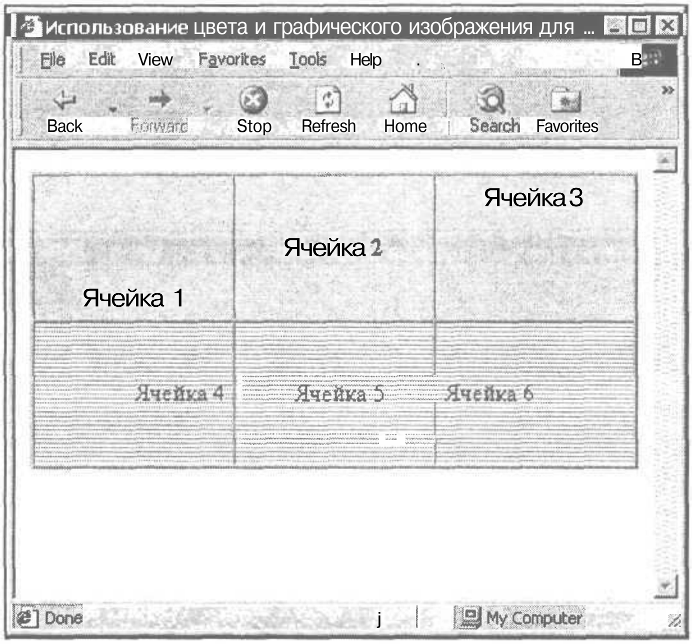


Рис. 4.7. Использование цвета и графического изображения для заднего фона ячеек

Параметры WIDTH и HEIGHT

Это параметры ширины и высоты либо отдельных ячеек, либо целых рядов таблицы. Правила указания аналогичны этим же параметрам тега <TABLE>. Используются для <TR>, <TD> и <тн>.

Параметр *NOWRAP*

Параметр NOWRAP запрещает принудительный перенос строки в ячейке или табличном ряде. Не рекомендуется использовать данный параметр во всех ячейках, т. к. это может сильно понизить уровнь масштабируемости таблицы (при условии, что значение ширины и/или высоты таблицы указано в процентах). Параметр NOWRAP применим в <TR>, <TD> и <тн>.

Параметры *COLSPAN* и *ROWSPAN*

Иногда при построении таблиц возникает необходимость в объединении нескольких соседних ячеек в одну. Для этого используются параметры COLSPAN и ROWSPAN.

COLSPAN показывает, на сколько ячеек по горизонтали следует расширить <TD> или <тн>, в котором указан данный параметр. ROWSPAN делает то же

самое, но увеличивает область <TD> или <тн> по вертикали (рис. 4.8, листинг 4.4).

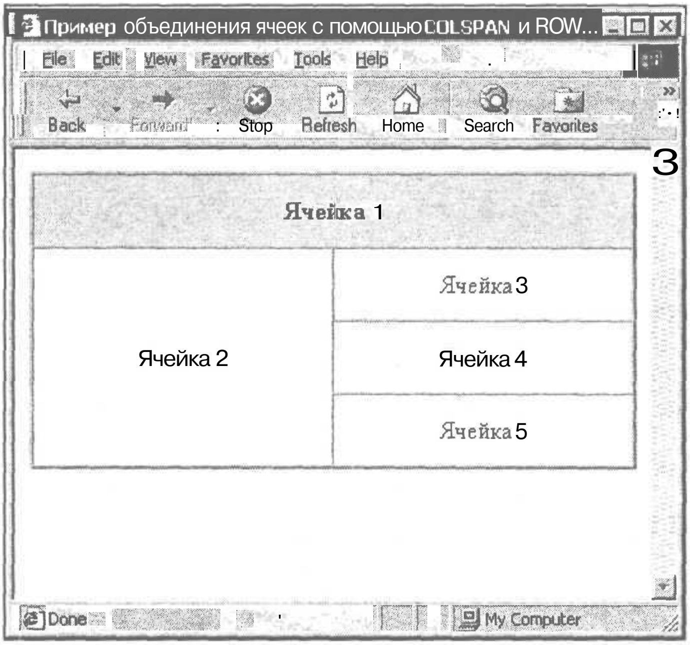


Рис. 4.8. Пример объединения ячеек с помощью COLSPAN и ROWSPAN

Листинг 4.4. Пример объединения ячеек с помощью COLSPAN и ROWSPAN

```
<HTML>
  <HEAD>
    <TITLE>Пример объединения ячеек с помощью COLSPAN и ROWSPAN</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue" LEFTMARGIN="40" RIGHTMARGIN="40" MARGINWIDTH="40">
<TABLE ALIGN="center" BORDER="2" CELLSPACING="0" CELLPADDING="5"
WIDTH="100%" HEIGHT="200">
<TR ALIGN="center" BGCOLOR="#CECECE">
<TH COLSPAN="2">Ячейка 1</TH>
</TR>
<TR>
<TD ALIGN="center" ROWSPAN="3">Ячейка 2</TD>
<TD ALIGN="center">Ячейка 3</TD>
```

```
</TR>
<TR>
<TD ALIGN="center">Ячейка 4</TD>
</TR>
<TR>
<TD ALIGN="center">Ячейка 5</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

При использовании параметров COLSPAN и ROWSPAN особое внимание следует уделять корректному объединению соседних ячеек таблицы, а также своевременному указанию закрывающих тегов ячеек и рядов. Возникновение ошибки может привести к нарушению структуры таблицы, "заползанию" одних ячеек на другие, перекрытию текста и даже невозможности отобразить таблицу в браузере.

Параметры COLSPAN и ROWSPAN используются только в тегах ячейки <TD> и <тн>.

Перечень параметров тегов <TR>, <TD> и <тн> приведен в табл. 4.2.

Таблица 4.2. Параметры тегов <TR>, <TD> и <тн>

Параметр	Функция	**Применение**
ALIGN	Выравнивание содержимого ячейки или ряда по горизонтали	<TR>, <TD>, <TH>
VALIGN	Выравнивание содержимого ячейки или ряда по вертикали	<TR>, <TD>, <TH>
WIDTH	Определение ширины ячейки или ряда	<TR>, <TD>, <TH>
HEIGHT	Определение высоты ячейки или ряда	<TR>, <TD>, <TH>
BGCOLOR	Указание цвета для фона ячейки или ряда	<TR>, <TD>, <TH>
BACKGROUND	Указание рисунка для фона ячейки	<TD>, <TH>
NOWRAP	Запрет принудительного переноса строки в ячейке или ряду	<TR>, <TD>, <TH>
COLSPAN	Объединение соседних ячеек по горизонтали	<TD>, <TH>
ROWSPAN	Объединение соседних ячеек по вертикали	<TD>, <TH>

Нестандартное представление таблиц

Этот подраздел главы посвящен некоторым специфическим особенностям построения таблиц, поддержка которых присутствует только в браузере Internet Explorer, а также различным нестандартным способам работы с таблицами.

Пустые ячейки

В процессе построения таблиц может возникнуть необходимость оставления одной или более ячеек незаполненными. Однако банальное отсутствие каких-либо символов внутри тегов <TD> и <тн> в зависимости от браузера может привести к самым разнообразным вариантам визуального отображения таблицы на экране. Пример отображения пустых ячеек таблицы различными браузерами приведен в листинге 4.5. Internet Explorer и Netscape не прорисовывают внутреннюю структуру пустых ячеек, а Opera к тому же оттеняет правую границу ячеек с информацией, размещенных слева от пустых табличных ячеек (рис. 4.9—4.11)

Листинг 4.5. Пример отображения пустых ячеек таблицы различными браузерами

```
<HTML>
  <HEAD>
    <TITLE>Пример отображения пустых ячеек таблицы различными браузера-
ми</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue" LEFTMARGIN="40" RIGHTMARGIN="40" MARGINWIDTH="40">
<TABLE ALIGN="center" BORDER="1" CELLSPACING="0" CELLPADDING="5"
WIDTH="100%" HEIGHT="200">
<TR ALIGN="center">
<TD>Заполненная ячейка</TD>
<TDX/TD>
<TD>Заполненная ячейка</TD>
</TR>
<TR ALIGN="center">
<TD>Заполненная ячейка</TD>
<TDX/TD>
<TD>Заполненная ячейка</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

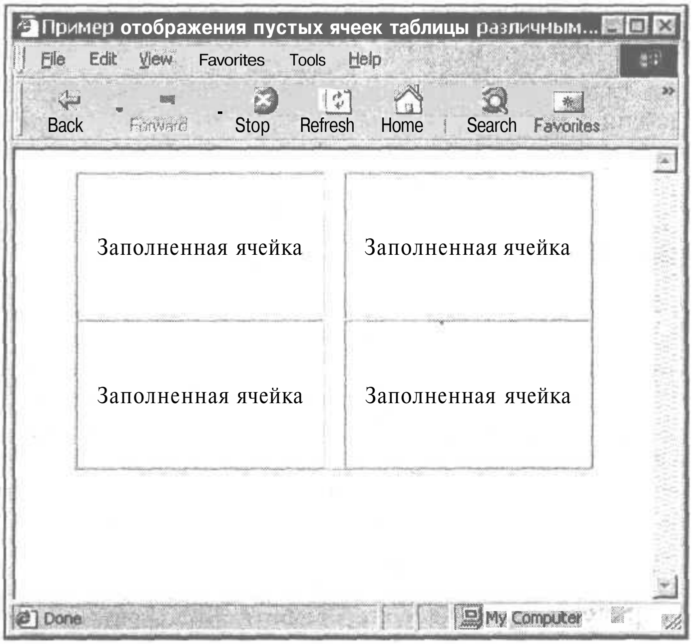


Рис. 4.9. Пример отображения пустых ячеек таблицы браузером Internet Explorer

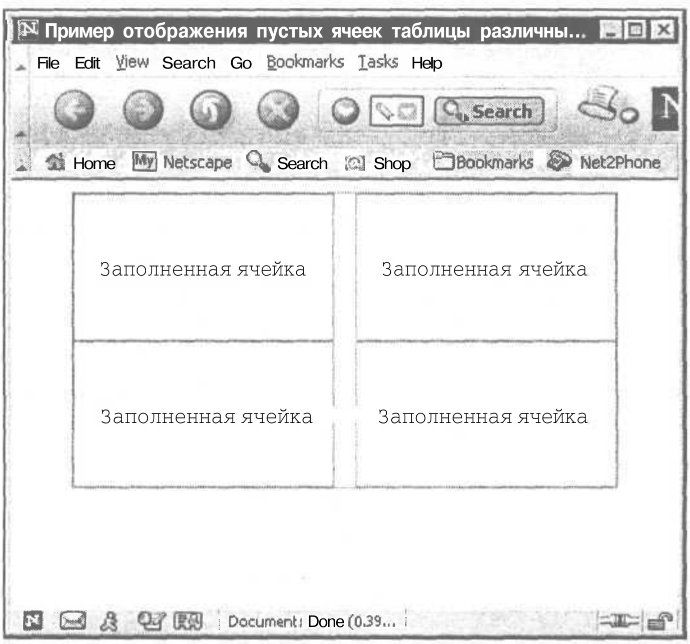


Рис. 4.10. Пример отображения пустых ячеек таблицы браузером Netscape

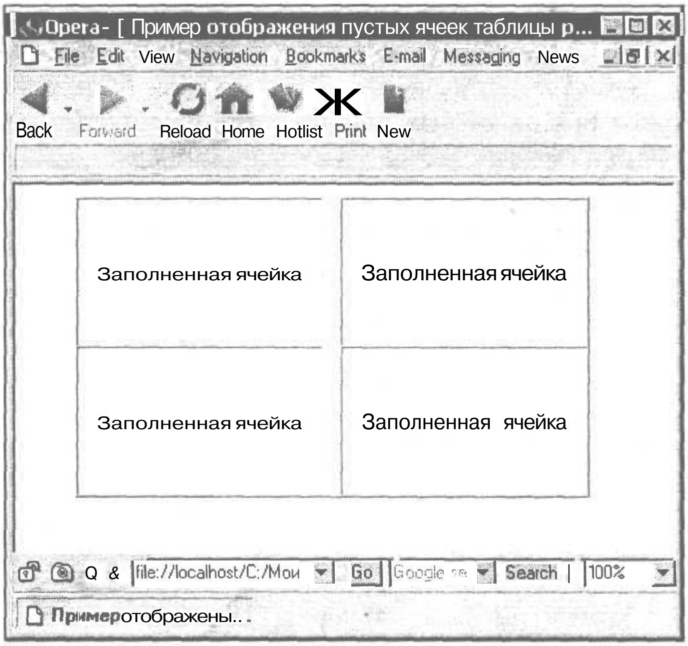


Рис. 4.11. Пример отображения пустых ячеек таблицы браузером Opera

Чтобы избежать различий в отображении таблиц, внутри тегов ячеек <TD> и <тн> следует размещать не видимые на экране данные, позволяющие браузерам не отображать такие ячейки как пустые. Подобные данные могут быть следующих типов:

- *теги структурного форматирования.* В данном случае под этим понятием подразумевается тег принудительного переноса строки
 и конструкция запрета такого переноса . При использовании этих элементов следует помнить, что размер пустой (в визуальном плане) ячейки будет зависеть от свойств шрифта, заданных для HTML-документа в целом или же предопределенных для конкретного блока (<SPAN>, <DIV>) или ячейки (<TD>, <TH>);
- *цветные текстовые блоки.* Разместив в ячейке, которая должна визуально казаться пустой, текст небольшого объема цветом, аналогичным значению параметра BGCOLOR тега <TD> или <тн>, при просмотре такого документа в браузере мы получим "пустую" табличную ячейку;
- *прозрачные графические распорки.* Это способ отображения пустых ячеек, практически не зависящий от модели браузера и свойств шрифта. Заключается в размещении прозрачного графического изображения нужных размеров (распорки, спейсера) в пределах табличной ячейки. Прозрачная распорка (в формате GIF) невидима на странице и очень гибка

в масштабировании. К примеру, невидимый спейсер исходного размера 10 x 10 пикселов при необходимости можно искусственно увеличить или уменьшить путем переназначения его параметров WIDTH и HEIGHT. Пример использования различных типов отображения пустых ячеек таблицы приведен соответственно на рис. 4.12 и в листинге 4.6.

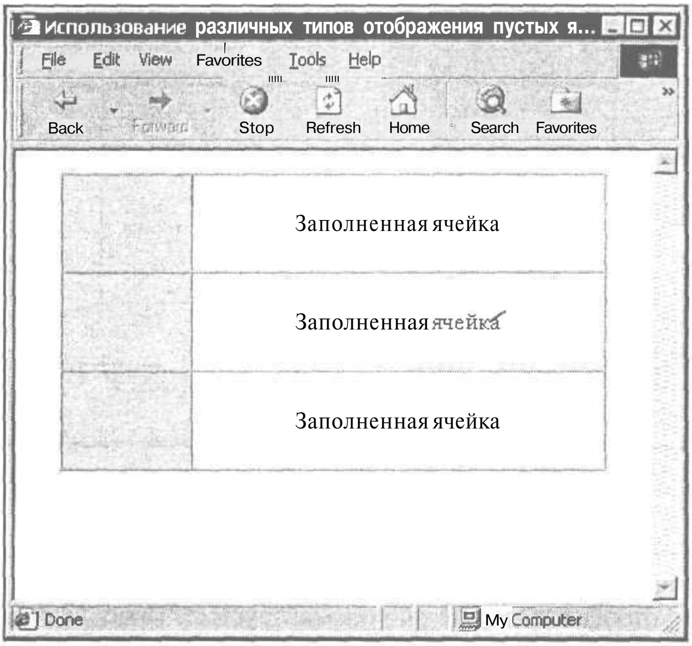


Рис. 4.12. Использование различных типов отображения пустых ячеек таблицы

Листинг 4.6. Использование различных типов отображения пустых ячеек таблицы

```
<HTML>
  <HEAD>
    <TITLE>Использование различных типов отображения пустых ячеек табли-
цы</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<TABLE ALIGN="center" BORDER="1" CELLSPACING="0" CELLPADDING="5"
WIDTH="90%" HEIGHT="200">
<TR ALIGN="center">
<TD BGCOLOR="#CECECE"> </TD>
```

```
<TD>Заполненная ячейка</TD>
</TR>
<TR ALIGN="center">
<TD BGCOLOR="#CECECE"><FONT COLOR="#CECECE">текст</FONT></TD>
<TD>Заполненная ячейка</TD>
</TR>
<TR ALIGN="center">
<TD BGCOLOR="#CECECE"><IMG SRC="spacer.gif" WIDTH="10" HEIGHT="10"
ALT=""></TD>
<TD>Заполненная ячейка</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Группировка данных

При построении таблиц мы можем легко задать единый тип выравнивания для отдельной ячейки и даже целого табличного ряда. Однако гораздо чаще может возникнуть необходимость в едином выравнивании информационных данных конкретного столбца. Столбец в таблице — это последовательность ячеек, располагающихся в разных рядах.

Стандартными средствами HTML нам пришлось бы задавать один и тот же тип выравнивания для отдельно взятой ячейки, формирующей нужный столбец:

```
<TABLE>
<TR>
<TD ALIGN="right">Ячейка 1 с выравниванием вправо</TD>
<TD ALIGN="center">Ячейка 1 с выравниванием по центру</TD>
</TR>
<TR>
<TD ALIGN="right">Ячейка 2 с выравниванием вправо</TD>
<TD ALIGN="center">Ячейка 2 с выравниванием по центру</TD>
</TR>
</TABLE>
```

При работе с браузером Internet Explorer, труд разработчика электронного документа может быть значительно облегчен за счет таких тегов, как <COL> и <COLGROUP>.

Теги <COL> и <COLGROUP> предназначены для определения свойств отображения табличных данных, сгруппированных по конкретному признаку. Они

оба могут иметь параметр SPAN, задающий количество соседних столбцов, и параметр ALIGN, устанавливающий единый тип выравнивания для выбранного столбца (возможные значения: по левому краю, по правому краю, по центру; формат записи аналогичен тегам <TD> и <TH>).

Помимо этого тег <COLGROUP> может содержать дополнительный параметр вертикального выравнивания данных — VALIGN (возможные значения: по верхнему краю, по нижнему краю, по середине; формат записи аналогичен тегам <TD> и <TH>).

Разница между тегами <COL> и <COLGROUP> заключается в условном объединении (группировке) взаимосвязанных данных отдельно взятого столбца таблицы. Пример группировки данных таблицы приведен соответственно на рис. 4.13 и в листинге 4.7.

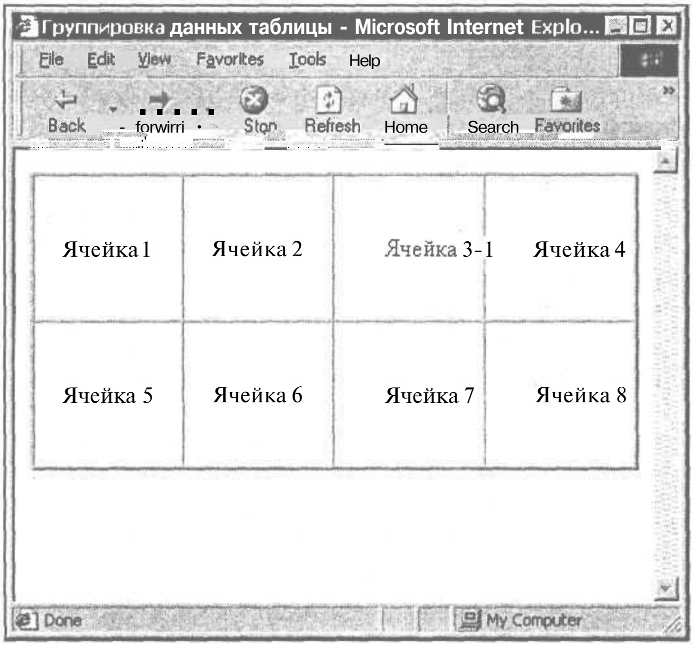


Рис. 4.13. Группировка данных таблицы

Листинг 4.7. Группировка данных таблицы

```
<HTML>
  <HEAD>
    <TITLE>Группировка данных таблицы</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
```

```
<TABLE ALIGN="center" BORDER="2" CELLSPACING="0" CELLPADDING="5"
WIDTH="100%" HEIGHT="200">
<COLGROUP ALIGN="center" SPAN="2">
<COLGROUP ALIGN="right" SPAN="2">
<TR>
<TD>Ячейка 1</TD><TD>Ячейка 2</TD><TD>Ячейка 3</TD><TD>Ячейка 4</TD>
</TR>
<TR>
<TD>Ячейка 5</TD><TD>Ячейка 6</TD><TD>Ячейка 7</TD><TD>Ячейка 8</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Из рисунка 4.13 видно, что таблица состоит из четырех столбцов, в которых данные сгруппированы по заданному признаку. Первые два столбца имеют тип выравнивания по центру, последние два — по правому краю.

Колонтитулы таблицы

Браузер Internet Explorer позволяет использовать дополнительные теги структурирования табличных данных, а именно - <THEAD>, <TBODY> и <TFOOT>. Эти теги предназначены для создания колонтитулов таблицы различных уровней (соответственно для верхнего, основного (содержательного) и нижнего уровней таблицы).

Теги верхнего и нижнего колонтитулов <THEAD> и <TFOOT> могут быть использованы в структуре таблицы лишь единожды, причем для них необязательно наличие закрывающих тегов.

Тег основного колонтитула <TBODY> может встречаться неоднократно в пределах одной таблицы, однако требует своего закрывающего тега.

Верхний и нижний колонтитулы функционально очерчивают логические заголовки соответствующего уровня и применимы, в основном, в больших таблицах, не помещающихся в пределах одной страницы электронного документа.

Основные колонтитулы выполняют функцию, аналогичную тегам группировки `<COL>` и `<COLGROUP>`, расставляя логические метки по ходу изложения основной содержательной части таблицы.

Прорисовка структуры таблицы

Еще одной замечательной возможностью нестандартного представления таблиц, работающей только в браузере Internet Explorer, является прорисовка внутренней структуры таблицы, а именно — рамок и линеек. Другими

словами, возможно оформить таблицу таким образом, что от рамки останутся только верхняя и нижняя границы, а между ячейками останется только вертикальная линейка.

За изменение свойств рамки отвечает параметр FRAME, а с помощью параметра RULES варьируется внешний вид линеек таблицы (оба параметра применимы внутри тега <TABLE>). Пример нестандартных возможностей прорисовки структуры таблицы приведен соответственно в листинге 4.8 и на рис. 4.14.

Листинг 4.8. Нестандартные возможности прорисовки структуры таблицы

```
<HTML>
  <HEAD>
    <TITLE>Нестандартные возможности прорисовки структуры таблицы</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<TABLE ALIGN="center" BORDER="1" CELLSPACING="0" CELLPADDING="4"
WIDTH="100%" FRAME="hsides" RULES="rows">
<CAPTION><B>Таблица 1</BX/CAPTION>
<TR>
<TD>Ячейка 1</TD><TD>Ячейка 2</TD><TD>Ячейка 3</TD><TD>Ячейка 4</TD>
</TR>
<TR>
<TD>Ячейка 5</TD><TD>Ячейка 6</TD><TD>Ячейка 7</TD><TD>Ячейка 8</TD>
</TR>
</TABLE>
<TABLE ALIGN="center" BORDER="1" CELLSPACING="0" CELLPADDING="4"
WIDTH="100%" FRAME="box" RULES="groups">
<CAPTION><B>Таблица 2</B></CAPTION>
<TR>
<TD>Ячейка 1</TD><TD>Ячейка 2</TD><TD>Ячейка 3</TD><TD>Ячейка 4</TD>
</TR>
<TR>
<TD>Ячейка 5</TD><TD>Ячейка 6</TD><TD>Ячейка 7</TD><TD>Ячейка 8</TD>
</TR>
</TABLE>
<TABLE ALIGN="center" BORDER="1" CELLSPACING="0" CELLPADDING="4"
WIDTH="100%" FRAME="above" RULES="all">
<CAPTION><B>Таблица 3</BX/CAPTION>
```

```
<TR>
<TD>Ячейка 1</TD><TD>Ячейка 2</TD><TD>Ячейка 3</TD><TD>Ячейка 4</TD>
</TR>
<TR>
<TD>Ячейка 5</TD><TD>Ячейка 6</TD><TD>Ячейка 7</TD><TD>Ячейка 8</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```



Рис. 4.14. Нестандартные возможности прорисовки структуры таблицы

Возможные значения параметров FRAME и RULES приведены соответственно в табл. 4.3 и 4.4.

Таблица 4.3. *Возможные значения параметра FRAME*

Значение	Функция
BOX	Рамка с четырех сторон
BORDER	Рамка с четырех сторон
ABOVE	Рамка только сверху
BELOW	Рамка только снизу

Таблица 4.3 (окончание)

Значение	Функция
HSIDES	Верхняя и нижняя части рамки
VSIDES	Левая и правая части рамки
LHS	Только левая часть рамки
RHS	Только правая часть рамки
VOID	Нет рамок

Таблица 4.4. *Возможные значения параметра RULES*

Значение	Функция
ALL	Отображение линейки целиком
GROUPS	Часть линейки, разделяющая сгруппированные данные
COLS	Часть линейки, разделяющая столбцы
ROWS	Часть линейки, разделяющая строки
NONE	Отсутствие линейки

Вложенные таблицы

Одной из замечательных особенностей HTML-таблиц по праву считается поддержка многоуровневой вложенности. Другими словами, одна таблица может включать другую, та, в свою очередь, еще одну и т. д.

Преимущества вложенных таблиц

Особенность вложенных таблиц, в отличие от других способов представления данных в электронном документе, позволяет более точно размещать отдельные элементы страницы относительно друг друга и относительно границ самого документа, отображаемого браузером.

Например, два разнородных блока текста и нумерованный список, размещенные внутри тега <BODY>, невозможно разместить на одном уровне, а тем более на одном уровне со смещением в какую-либо сторону. Использование таблиц с легкостью решает эту проблему, позволяя располагать различные элементы и их комбинации в разных местах документа посредством видимых и невидимых ячеек рядов таблицы.

Вот почему в последнее время преобладающее большинство HTML-документов создается на основе таблиц, где в качестве несущей основы берется таблица с невидимыми краями, содержащая вложенные таблицы с разным оформлением, отличающимися значениями параметров.

Подводя итог сказанному, можно выделить следующие преимущества вложенных таблиц:

- гибкая масштабируемость структуры электронного документа в целом;
- широкие возможности позиционирования отдельных элементов страницы;
- многоуровневое представление разнородных информационных данных;
- расширенные оформительские возможности;
- поддержка популярными браузерами.

Пример вложенных таблиц

Правила построения вложенных таблиц ничем не отличаются от создания таблиц одного уровня — используются те же теги и параметры, задаются те же свойства и значения.

Единственное, о чем нельзя забывать в ходе создания сложных вложенных таблиц, это:

- каждая таблица последующего уровня размещается внутри тега-контейнера <TD> или <TH> таблицы предыдущего уровня;
- вложенная таблица не может быть создана за пределами вышеназванных тегов ячейки таблицы;
- таблица одного уровня может содержать любое количество вложенных таблиц другого уровня, идущих друг за другом в пределах тега ячейки таблицы верхнего уровня;
- количество тегов таблиц всех уровней должно соответствовать количеству закрывающих тегов этих же таблиц.

Один из вариантов использования многоуровневых вложенных таблиц приведен соответственно в листинге 4.9 и на рис. 4.15.

Листинг 4.9. Пример использования многоуровневых вложенных таблиц

```
<HTML>
  <HEAD>
    <TITLE>Пример использования многоуровневых вложенных таблиц</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H2 ALIGN="center">ПОСТРОЕНИЕ ВЛОЖЕННЫХ ТАБЛИЦ</H2>
<!-- Основная несущая таблица. Начало -->
<TABLE ALIGN="center" BORDER="0" CELLSPACING="0" CELLPADDING="2"
WIDTH="100%">
```

```
<TR>
<TD VALIGN="top" WIDTH="50%">
<!-- Левая таблица с текстом. Начало -->
<TABLE ALIGN="center" BORDER="1" CELLSPACING="3" CELLPADDING="5"
WIDTH="100%">
<TR>
<TH BGCOLOR="gray"><FONT COLOR="white">Вложенные таблицы</FONT></TH>
</TR>
<TR>
<TD VALIGN="top">
<P ALIGN="justify">
Одной из замечательных особенностей HTML-таблиц по праву считается под-
держка многоуровневой вложенности.
<BRXBR>
Другими словами, одна таблица может включать другую, та, в свою очередь,
еще одну и т. д.
<BRXBR>
Эта особенность, в отличие от других способов представления данных в
электронном документе, позволяет более точно размещать отдельные элементы
страницы относительно друг друга и границ самого документа, отображаемого
браузером.
</P>
</TD>
</TR>
</TABLE>
<!-- Левая таблица с текстом. Окончание -->
</TD>
<TD VALIGN="top" WIDTH="50%">
<!-- Правая таблица со списком. Начало -->
<TABLE ALIGN="center" BORDER="1" CELLSPACING="3" CELLPADDING="5"
WIDTH="100%">
<TR>
<TH BGCOLOR="#000000"><FONT COLOR="white">Преимущества вложенных таб-
лиц</FONT></TH>
</TR>
<TR>
<TD VALIGN="top">
<OL TYPE="1">
    <LI>Гибкая масштабируемость структуры электронного документа в це-
лом<BR><BR> •
```

```
    <LI>Широкие возможности позиционирования отдельных элементов страницы<BR><BR>
    <LI>Многоуровневое представление разнородных информационных данных<BR><BR>
    <LI>Расширенные оформительские возможности<BR><BR>
    <LI>Поддержка популярными браузерами
</OL>
</TD>
</TR>
</TABLE>
<!-- Правая таблица со списком. Окончание -->
</TD>
</TR>
</TABLE>
<!-- Основная несущая таблица. Окончание -->
</BODY>
</HTML>
```

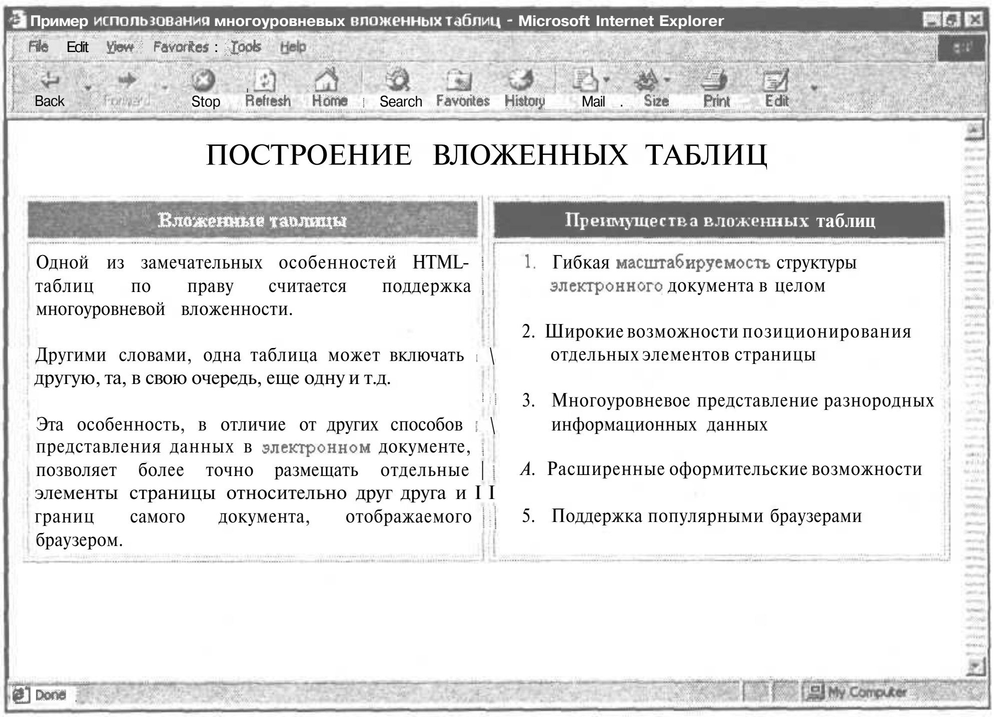


Рис. 4.15. Пример использования многоуровневых вложенных таблиц

Как видно из листинга 4.9, электронный документ состоит из трех таблиц: первая является основной (несущей) и имеет невидимые рамки; две других расположены в ячейках основной таблицы и содержат блок текста и нумерованный список, соответственно.

Для удобства восприятия HTML-кода были использованы комментарии, обособляющие структуру каждой из трех таблиц. Ширина ячеек несущей таблицы — по 50%, что позволяет даже при изменении размеров окна браузера разделять информацию на две равные части.

Для закрепления пройденного материала все описанные теги и параметры, используемые при построении HTML-таблиц, приведены в табл. 4.5.

Таблица 4.5. *Теги и параметры, используемые для построения таблиц*

Тег	Описание	Параметры
TABLE	Основной тег построения структуры таблицы. Определяет наличие и цвет рамки, отступы между границами ячеек и соседними ячейками, тип выравнивания, ширину и высоту таблицы. Задает свойства прорисовки рамок и линеек таблицы	BORDER BORDERCOLOR CELLSPACING CELLPADDING ALIGN WIDTH HEIGHT FRAME RULES
CAPTION	Тег заголовка таблицы. Задает тип горизонтального и вертикального выравнивания заголовка таблицы	ALIGN VALIGN
TR	Тег ряда таблицы. Устанавливает следующие свойства для ряда таблицы: тип вертикального и горизонтального выравнивания, ширину и высоту, цвет фона, запрет переноса строки	ALIGN VALIGN WIDTH HEIGHT BGCOLOR NOWRAP
TD, TH	Теги ячейки таблицы. Устанавливает следующие свойства для ячейки таблицы: тип вертикального и горизонтального выравнивания, ширину и высоту, цвет и графическое изображение фона, запрет переноса строки, обьединение соседних ячеек в одну (по горизонтали и вертикали)	ALIGN VALIGN WIDTH HEIGHT BGCOLOR BACKGROUND NOWRAP ROWSPAN COLSPAN

***Таблица 4.5** (окончание)*

Тег	Описание	Параметры
COL	Тег группировки табличных данных. Задает количество соседних столбцов и единый тип горизонтального выравнивания для всех столбцов	SPAN ALIGN
COLGROUP	Тег группировки табличных данных, осуществляющий логическое объединение столбцов с информацией. Задает количество соседних столбцов и единый тип горизонтального и вертикального выравнивания для всех столбцов	SPAN ALIGN VALIGN
THEAD	Тег верхнего колонтитула таблицы	—
TBODY	Тег основного (содержательного) колонтитула таблицы	—
TFOOT	Тег нижнего колонтитула таблицы	—

Глава 5

Добавляем графику

Роль графики

Как интересный журнал или проспект теряет в своей привлекательности без цветных иллюстраций, так любой HTML-документ кажется сухим и невзрачным без использования графики. Значение графических изображений в аспекте создания электронных документов нельзя переоценить – реклама и коммерческие предложения компаний и юридических лиц становятся более выразительными и яркими, иллюстрации и схемы способны превратить скучный перечень услуг или расценок в интересный информативный материал, любой художественный рассказ или произведение будет восприниматься легче и естественнее при наличии картинок или фотографий.

Однако всегда и во всем следует помнить о чувстве меры. HTML-документ, перенасыщенный иллюстративным материалом, будет неоправданно отвлекать внимание пользователя от истинного содержания страницы — информации. К тому же чрезмерное увлечение графикой влечет за собой увеличение времени загрузки электронного документа, что может сказаться на общей оценке посетителем вашего интернет-ресурса.

Также необходимо четко и внимательно отбирать графический материал для последующего размещения в HTML-документах, следить за его соответствием тематике вашего интернет-ресурса в целом и содержанию отдельной страницы в частности.

Следует помнить о том, что графика призвана привлечь внимание посетителя, заострить его интерес на конкретных моментах или формулировках, но ни в коем случае не должна отвлекать от основного содержания Web-сайта. Разумеется, данное утверждение имеет ряд оговорок и исключений из правил (например, в отношении компьютерных галерей и прочих ресурсов, где ставка сделана именно на графические изображения). И, тем не менее, в большинстве случаев необходимо строго следить за количеством графики на ваших HTML-документах.

Характеристика графических стандартов

Любая графическая информация может храниться в двух основных форматах — векторном и растровом.

Графический файл векторного формата состоит из отдельных математических данных, которые с помощью отрезков прямых, называемых векторами, отображают графический объект на экране. Преимуществом векторной графики является независимость качества изображения от масштаба рисунка, а минусом — недостаточная возможность для работы с фотографическими изображениями. Примером файлов векторного формата могут служить CDR (CorelDRAW), AI (Adobe Illustrator), FH (FreeHand), SVG (Scalable Vector Graphics) и др. Векторный формат распространен, в основном, в полиграфии, предпечатной подготовке высококачественных документов и т. д.

Отображение файла растрового формата основано на обработке минимальной единицы рабочей области экрана — точки (пиксела). Изменение размеров растровых изображений может существенно повлиять на их качество, т. к. масштабирование осуществляется без вмешательства каких-либо сложных математических операций. Наиболее распространенными растровыми форматами являются PSD (Photoshop Document), BMP (Bitmap Image) и др.

Форматом, избранным для демонстрации Web-графики в Интернете, стал растровый формат, поскольку относительно малый размер мониторов пользователей не позволяет выводить на экран изображения больших размеров.

На сегодняшний день для создания графических изображений, предназначенных для размещения в электронных документах, используются три основных стандарта: GIF, JPEG и PNG. Рассмотрим каждый из них немного подробнее.

GIF (Graphics Interchange Format)

Стандарт GIF был разработан компанией CompuServe Inc. для передачи графической информации в пределах определенных компьютерных сетей (разработка велась еще до появления Интернета).

Надо заметить, что в изображении этого стандарта количество цветов не превышает числа 256. Среди других характеристик стандарта GIF можно назвать возможность построчного чередования служебных заголовков блоков с данными файла, внесения комментариев, реакцию на пользовательские действия (перемещение курсора и т. д.), создание прозрачного фона и поддержку режима анимации кадров.

Две последние характеристики стандарта GIF сегодня являются основополагающими и повсеместно используемыми — от создания навигационных меню до интернет-рекламы. К сожалению, такое замечательное свойство, как реакция на манипуляции пользователя, не реализовано в интернет-технологиях, поскольку оно не поддерживается браузерами.

На сегодняшний день стандарт GIF по-прежнему является самым используемым и популярным в ходе разработки электронных документов благодаря следующим преимуществам:

- наличие специального алгоритма сжатия данных позволяет подвергать компрессии файлы GIF без изменения качества изображения;
- способность чередования кадров анимированного GIF-файла приводит к тому, что в пределах одного рекламного носителя возможно разместить большой объем информации;
- по сравнению со статичными изображениями анимированный GIF-объект привлекает большее внимание со стороны пользователей;
- поддержка прозрачности позволяет экономить на исходном размере файла.

JPEG (Joint Photographic Experts Group)

Как видно из расшифровки аббревиатуры стандарта JPEG, он был создан специальной группой экспертов в области фотографии и предназначался для хранения графических изображений с большой глубиной цвета.

Профиль деятельности разработчиков JPEG внес свой вклад в позиционирование стандарта — в основном он использовался (и по сей день используется) для передачи фотографических изображений. В стандарте применен специальный алгоритм компрессии данных — при повышении степени сжатия качество изображения ухудшается за счет изъятия "ненужной" информации (в отличие от алгоритма сжатия GIF, который позволяет производить подобную процедуру практически без потерь).

На сегодняшний день стандарт JPEG занимает второе место по популярности после GIF и используется для создания изображений, в композицию которых входят фотографии, сложные коллажи (компьютерный монтаж нескольких разнородных графических объектов), объекты, подвергнутые действию различных графических эффектов и фильтров.

PNG (Portable Network Graphics)

Стандарт разрабатывался с учетом особенностей Интернета. PNG вобрал в себя наиболее сильные стороны двух предыдущих стандартов и исключил их недостатки.

В стандарте реализованы следующие средства:

- прозрачный фон;
- построчное чередование;
- сжатие без потерь и др.

Однако PNG, несмотря на свои преимущества, не получил такого признания пользователями Интернета, как стандарты GIF и JPEG (одна из возможных причин — отсутствие поддержки анимации). В Сети можно встретить графические файлы с расширением png, но достаточно редко и только в специфических направлениях интернет-отрасли (например, работа программного модуля GD::Graph, автоматически генерирующего графики и диаграммы для систем статистики, рейтинга и пр.).

Вставка графики в HTML-документ

Прежде чем перейти к рассмотрению правил встраивания графики в HTML-документы, следует напомнить, что графические изображения также могут быть включены в основной раздел документа, описываемый тегом-контейнером <BODY>, а также в качестве фона ячеек таблицы:

```
<BODY BACKGROUND="bg.gif">
<TABLE>
<TR>
<TD BACKGROUND="bg2.jpg">Текст</TD>
</TR>
</TABLE>
```

Описание графики в HTML-документе

Для вставки графических изображений в HTML-документы используется специальный тег <IMG>, который не требует наличия закрывающего тега.

Параметры тега *<IMG>*

Рассмотрим существующие параметры этого тега.

Параметр *SRC*

Единственный параметр, который является обязательным для указания, — это SRC, который выполняет важную роль в графическом изображении на странице — он задает путь (относительный или абсолютный) к рисунку. Формат указания следующий:

```
<IMG SRC="picture.gif">
```

Следует заметить, что в данном случае браузер станет искать файл `"picture.gif"` в том же каталоге, где находится и HTML-документ, ссылающийся на этот рисунок. Обычно для графических изображений выделяется отдельная папка:

```
<IMG SRC="images/picture.gif">
```

Параметр *BORDER*

Параметр BORDER предназначен для отображения вокруг рисунка рамки определенной толщины (рис. 5.1). Толщина указывается в пикселах, по умолчанию (если параметр пропущен) рамка браузером не отображается:

```
<IMG SRC="images/clock.gif" BORDER="5">
```

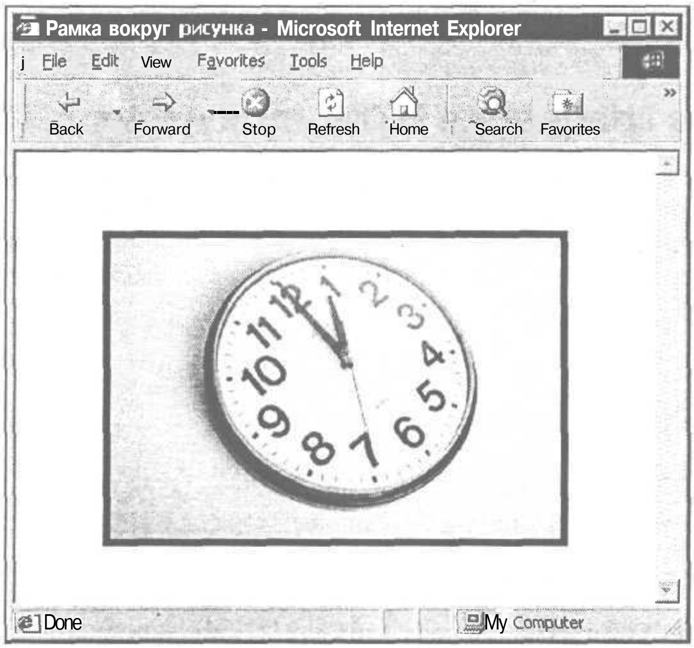


Рис. 5.1. Пример рамки вокруг рисунка

В случае если графическое изображение является гиперссылкой, браузер автоматически отобразит вокруг рисунка рамку толщиной в 1 пиксел (кроме этого, некоторые браузеры делают рамку определенного цвета, обычно синего).

Поэтому, если никакой необходимости в рамке вокруг графического указателя ссылки нет, следует дать браузеру соответствующую инструкцию:

```
<A HREF="clock.html"><IMG SRC="images/clock.gif" BORDER="0"></A>
```

Параметры *WIDTH* и *HEIGHT*

По аналогии с другими HTML-элементами (табличные ячейки, горизонтальные разделители) параметры WIDTH и HEIGHT используются для определения размеров графического изображения, встраиваемого в электронный документ (ширина и высота, в пикселах):

```
<IMG SRC="images/clock.gif" BORDER="1" WIDTH="300" HEIGHT="201">
```

Использование этих параметров не обязательно, однако рекомендуется по двум основным причинам:

- во время загрузки изображения браузер сразу зарезервирует на странице столько места, сколько необходимо для отображения рисунка. Многие разработчики пренебрегают этим правилом, в результате чего при загрузке под рисунок изначально отводится слишком мало места, а потом страница начинает "скакать", поскольку браузер одновременно пытается в это пространство вместить реальные размеры файла;
- иногда размеры изображения слишком велики и для его просмотра "в полный рост" требуются полосы прокруток. В таком случае, если необходимо поместить рисунок в окне браузера, не прибегая к помощи полос прокруток, то можно в HTML-коде указать пропорционально уменьшенные размеры этого рисунка. Размещая на странице уменьшенный вариант рисунка, обязательно помните о необходимости пропорционального изменения размеров, в противном случае при нарушении соотношения этих размеров, изображение получится искаженным и неприглядным (рис. 5.2).

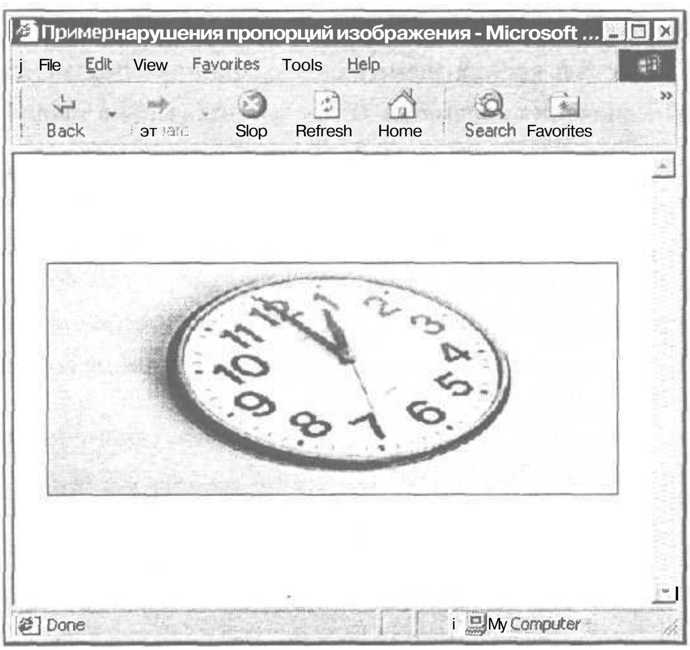


Рис. 5.2. Пример нарушения пропорций изображения

Корректно изменить пропорции рисунка можно следующими способами:

- указать в теге <IMG> измененным только один из параметров (либо WIDTH, либо HEIGHT). Браузер самостоятельно изменит второй параметр, сохраняя пропорцию;

- открыть рисунок в любом растровом редакторе (например, Adobe Photoshop) и изменить пропорции программным образом.

Что касается пропорционального увеличения графических изображений в HTML-документах, то такой подход не рекомендуется по причине особенностей самого растрового формата (ухудшение качества изображения при увеличении его размеров). Однако если степень увеличения незначительна (в среднем на 5—10 пикселов), как правило, визуально качество остается неизменным.

В завершение разговора о параметрах WIDTH и HEIGHT следует сказать, что допустимо указание размеров графического изображения в процентах. Этот формат записи дает команду браузеру растянуть или сузить рисунок в соответствии с размерами окна страницы. Но такой подход не рекомендован и может применяться только в исключительных случаях и только в отношении однородных рисунков, поскольку некоторые браузеры либо не поддерживают подобное обозначение размеров, либо выводят рисунки некорректно.

Параметр *ALIGN*

Параметр ALIGN предназначен для определения типа выравнивания рисунка относительно текста и прочих элементов электронного документа.

Существующие значения параметра ALIGN представлены в табл. 5.1.

Таблица 5.1. *Значения параметра ALIGN*

Значение	Функция
LEFT	Текст обтекает рисунок по правому краю (изображение слева)
RIGHT	Текст обтекает рисунок по левому краю (изображение справа)
TOP	Верхний край рисунка выравнивается по самому высокому элементу строки
TEXTTOP	Верхний край рисунка выравнивается по самому высокому текстовому элементу строки
MIDDLE	Выравнивание середины рисунка по базовой линии строки
ABSMIDDLE	Выравнивание середины рисунка по середине строки
BOTTOM	Выравнивание нижнего края рисунка по базовой линии строки
BASELINE	Выравнивание нижнего края рисунка по базовой линии строки
ABSBOTTOM	Выравнивание нижнего края рисунка по нижнему краю строки

Следует заметить, что некоторые популярные браузеры (например, Netscape) интерпретируют значения ABSMIDDLE и ABSBOTTOM как, соответственно, MIDDLE и BOTTOM. Internet Explorer корректно отображает все приведенные в табл. 5.1 значения.

Основное же различие между значениями нижнего края и базовой линии заключается в особенностях некоторых букв алфавита и прочих символов. Нижний край — это крайняя точка текстового блока, а базовая линия проходит по нижней части строки текста без учета фрагментов текста, выходящих за строку. Другими словами, при наличии в тексте букв "у", "р", "q", "g" и т. п. нижняя часть этих символов будет определяться значением ABSBOTTOM, а часть букв на уровне запятой в предложении — значением BOTTOM или BASELINE. Примеры выравнивания рисунков приведены соответственно в листинге 5.1 и на рис. 5.3.

Листинг 5.1. Примеры выравнивания рисунков

```
<HTML>
  <HEAD>
    <TITLE>Примеры выравнивания рисунков</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<IMG SRC="brush.jpg" ALIGN="left" BORDER="1" WIDTH="100"
HEIGHT="67"><B>Выравнивание текста по правому краю рисун-
ка</B><BR>Стандарт GIF был разработан компанией CompuServe Inc. для
передачи графической информации в пределах определенных компьютерных
сетей (разработка велась еще до появления Интернета). Надо заметить, что
в изображении этого стандарта количество цветов ограничивается всего лишь
256-ю, однако нельзя сказать, чтобы это являлось большим недостатком.
Среди других черт стандарта GIF можно назвать возможность построчного
чередования, внесение комментариев, реакцию на пользовательские действия,
создание прозрачного фона и поддержку режима анимации кадров.
<HR>
<IMG SRC="brush.jpg" ALIGN="right" BORDER="1" WIDTH="100"
HEIGHT="67"><B>Выравнивание текста по левому краю рисунка</B><BR>Стандарт
GIF был разработан компанией CompuServe Inc. для передачи графической
информации в пределах определенных компьютерных сетей (разработка велась
еще до появления Интернета). Надо заметить, что в изображении этого
стандарта количество цветов ограничивается всего лишь 256-ю, однако
нельзя сказать, чтобы это являлось большим недостатком. Среди других черт
стандарта GIF можно назвать возможность построчного чередования, внесение
комментариев, реакцию на пользовательские действия, создание прозрачного
фона и поддержку режима анимации кадров.
<HR>
<B>Выравнивание этого предложения <IMG SRC="brush.jpg" ALIGN="top"
BORDER="1" WIDTH="100" HEIGHT="67"> по верхнему краю рисунка</B>
</BODY>
</HTML>
```

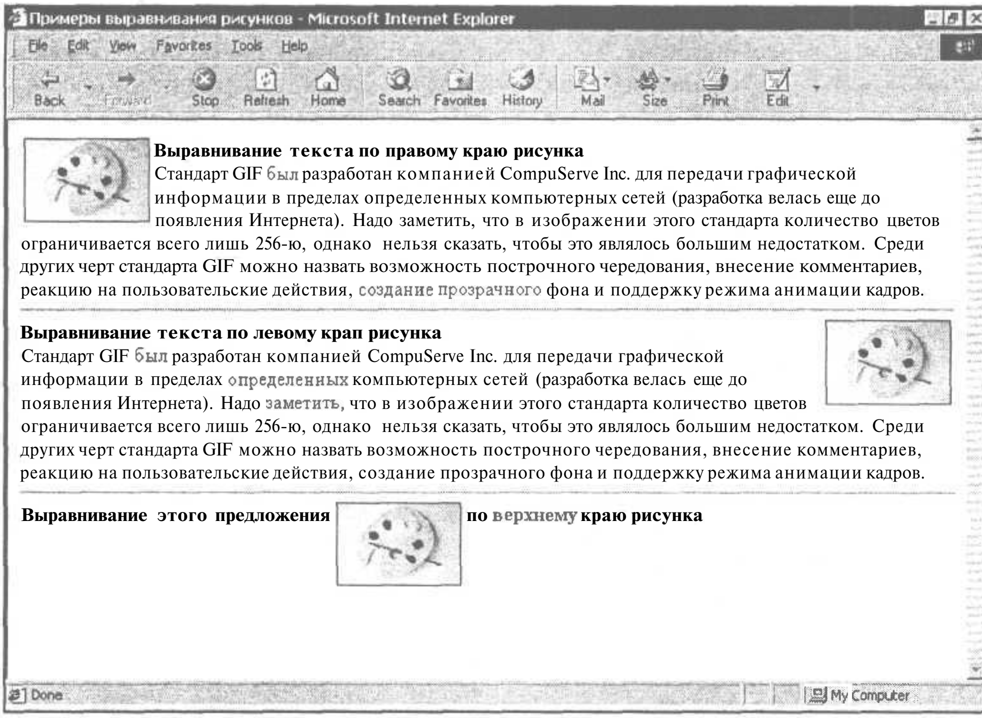


Рис. 5.3. Примеры выравнивания рисунков

Как видно из примеров, все типы выравнивания условно можно разделить на две группы:

- выравнивание относительно края окна документа (значения LEFT и RIGHT);
- выравнивание относительно текущей строки текста (все остальные значения).

Параметры *HSPACE* и *VSPACE*

Параметры HSPACE и VSPACE определяют размер горизонтального и вертикального отступов от встраиваемого изображения до текста. Формат записи значений — в пикселах. По умолчанию значения обоих типов отступа равны нулю (листинг 5.2, рис. 5.4).

Листинг 5.2. Отступы между рисунком и текстом

```
<HTML>
  <HEAD>
    <TITLE>Отступы между рисунком и текстом</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
```

```
<IMG SRC="brush.jpg" ALIGN="left" BORDER="1" WIDTH="100"
HEIGHT="67"><B>Отступов между текстом и изображением нет</B><BR>Стандарт
GIF был разработан компанией CompuServe Inc. для передачи графической
информации в пределах определенных компьютерных сетей (разработка велась
еще до появления Интернета). Надо заметить, что в изображении этого
стандарта количество цветов ограничивается всего лишь 256-ю, однако
нельзя сказать, чтобы это являлось большим недостатком. Среди других черт
стандарта GIF можно назвать возможность построчного чередования, внесение
комментариев, реакцию на пользовательские действия, создание прозрачного
фона и поддержку режима анимации кадров.

<HR>

<IMG SRC="brush.jpg" ALIGN="left" BORDER="1" WIDTH="100" HEIGHT="67"
HSPACE="20" VSPACE="20"><B>Отступ между текстом и изображением равен 20
пикселам</B><BR>Стандарт GIF был разработан компанией CompuServe Inc.
для передачи графической информации в пределах определенных компьютерных
сетей (разработка велась еще до появления Интернета). Надо заметить, что
в изображении этого стандарта количество цветов ограничивается всего лишь
256-ю, однако нельзя сказать, чтобы это являлось большим недостатком.
Среди других черт стандарта GIF можно назвать возможность построчного
чередования, внесение комментариев, реакцию на пользовательские действия,
создание прозрачного фона и поддержку режима анимации кадров.

</BODY>

</HTML>
```

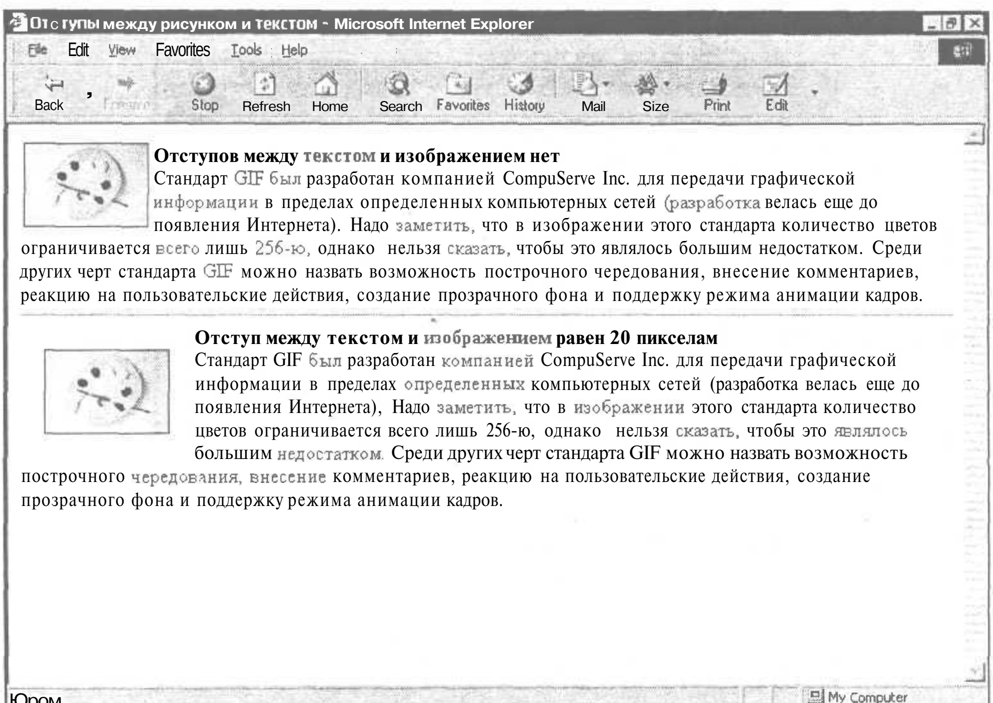


Рис. 5,4. Отступы между рисунком и текстом

Необходимо обратить внимание на то, что значение, указанное для параметров HSPACE и VSPACE, устанавливается с обеих сторон графического изображения.

Параметр *ALT*

Параметр альтернативного текста ALT играет существенную роль в процессе встраивания графики в электронные документы.

Изначальное его предназначение заключалось в возможности предоставления пользователям неграфических браузеров информации о рисунке, которая появляется при наведении на графический объект курсора мыши. Однако сегодня понятие "неграфического браузера" стало практически неактуальным, поэтому функция параметра ALT сильно расширилась и повсеместно используется для вывода смысловых подсказок для значимых графических элементов. Для не имеющих какой-либо смысловой нагрузки рисунков альтернативный текст указывать не следует.

Например, для рисунка `"services.gif"`, демонстрирующего образ услуг коммерческой компании, будет уместным добавить альтернативный текст "Наши услуги". Для прозрачной распорки, размещенной в пустой табличной ячейке, совершенно бессмысленно писать что-то вроде "Прозрачная графическая распорка".

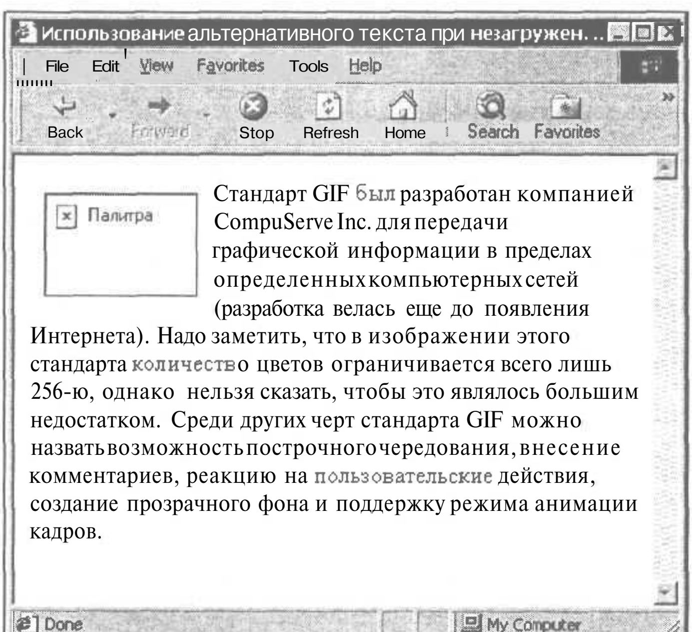


Рис. 5.5. Использование альтернативного текста при незагруженном изображении

Для незначимых графических изображений обычно задается следующая конструкция:

```
<IMG SRC="images/spacer.gif" WIDTH="5" HEIGHT="5" ALT="">
```

Таким образом, даже если ваш HTML-документ будут просматривать в текстовом редакторе (к примеру, Lynx), композиция страницы сохранится, а пользователю будет выведена информация о наличии рисунка.

Также указывать альтернативный текст ALT рекомендуется в случае, если изображение по каким-либо причинам не загрузилось. Подсказка, отображенная на месте рисунка, даст пользователю хотя бы минимальную информацию о незагруженном объекте (рис. 5.5).

Параметр *TITLE*

Данный параметр, по сути, выполняет функцию, аналогичную ALT, и предназначен для отображения подсказок к рисункам. Различие в том, что при незагруженном изображении параметр TITLE не предоставит пользователю никакой информации — для этого существует параметр альтернативного текста ALT.

Следует отметить, что последние версии некоторых браузеров (Opera, Netscape) не отображают значение поля ALT при наведении на изображение курсора мыши.

Для закрепления материала все рассмотренные параметры тега <IMG> приведены в табл. 5.2.

Таблица 5.2. *Параметры тега* <IMG>

Параметр	**Функция**
SRC	Указание пути к встраиваемому изображению
BORDER	Указание рамки рисунка
WIDTH	Определение ширины изображения
HEIGHT	Определение высоты изображения
ALIGN	Указание типа выравнивания рисунка
HSPACE	Определение горизонтального отступа между рисунком и текстом
VSPACE	Определение вертикального отступа между рисунком и текстом
ALT	Указание альтернативного текста
TITLE	Создание информационной подсказки к изображению

Карты-изображения

На многих HTML-документах сегодня с успехом используются так называемые карты-изображения (Imagemaps), которые представляют собой обычные графические файлы (как правило, стандарта GIF или JPEG) с привязанными к различным областям этого изображения гиперссылками.

Такие области описываются специальными координатами, в соответствии с которыми браузер переносит пользователя на нужную страницу.

О правилах конфигурации карт-изображений будет рассказано чуть позже, а сейчас рассмотрим преимущества и недостатки карт-изображений как средства навигации (перемещения) по HTML-документам.

Преимущества и недостатки карт-изображений

К основным преимуществам можно отнести следующие моменты:

- более удобного средства, чем карта-изображение, для создания сложных навигационных меню (в особенности географических, топографических и прочих карт) не найти. Процесс создания и пространственного размещения на странице нескольких десятков кнопок для обозначения, например, всех областей Российской Федерации, чрезвычайно сложен и потребует больших временных затрат;
- использование карты-изображения в качестве навигационных меню на каждой странице интернет-проекта может существенно сократить время загрузки электронных документов и сэкономить место на Web-сервере;
- для использования карты-изображения потребуется изготовить всего один рисунок;
- использование карт-изображений позволит разработчику HTML-документов реализовать самые смелые дизайнерские задумки. Можно создавать графические объекты любой сложности и формы, не задумываясь об их пространственном размещении на странице, что способно придать интернет-ресурсу оригинальность и сделать его более запоминающимся для посетителей.

Однако без некоторых недостатков также не обошлось:

- графические заготовки для карт-изображений, имеющих большой размер файла, могут заметно увеличить время загрузки электронных документов по сравнению с обычными текстовыми ссылками;
- для посетителей, которые экономят свое время пребывания в Интернете, переход к HTML-документам по ссылкам, указанным в конфигурации карты-изображения, не позволяет отслеживать страницы, на которых они уже побывали, поскольку гиперссылки карт-изображений не изменяют цвет после посещения их пользователем;

- при наведении курсора мыши на определенную активную область карты-изображения в серверном варианте, в статусной строке браузера отображается не адрес электронного документа, а координаты области, которые хранятся в конфигурационном файле на самом Web-сервере;
- если параллельно с картой-изображением не предусмотрено дублирующее текстовое меню, то посетители, которые по каким-либо причинам не могут загрузить графику или отключили ее в своем браузере, останутся не у дел.

Область применения

В принципе, карты-изображения можно применять в самых разнообразных областях компьютерных технологий. Наиболее распространенными из них являются:

- геоинформационные и картографические системы;
- баннерные рекламные сети и системы электронной коммерции;
- электронный и сотовый банкинг, платежные системы;
- игровые трехмерные и двумерные интернет-ресурсы;
- корпоративные серверы;
- интернет-ресурсы широкого профиля.

Как видно из списка, диапазон применения карт-изображений может охватывать практически все отрасли современных технологий.

Конфигурация карт-изображений

Прежде всего следует сказать, что карта-изображение не является принципиально новой технологией. Однако раньше их применение было осложнено рядом причин:

- ранние версии некоторых браузеров (например, Netscape Navigator 1.*x*) не поддерживали технологию карт-изображений;
- часть Web-серверов не позволяло использовать графику для создания карт-изображений вообще или предоставляло такую возможность только лишь профессиональным разработчикам или крупным компаниям;
- раньше был возможен единственный вариант реализации карт-изображений — серверный, который подразумевал управление переходом на соответствующий документ исключительно со стороны сервера.

Сегодня существует клиентский вариант карт-изображений, что во многом облегчает применение данной технологии, которую в настоящее время поддерживает большинство современных браузеров.

Сама карта-изображение представляет собой обыкновенный графический рисунок, а ее конфигурация определяется в виде значений координатных

5 Зак. 863

кривых, которые прописываются в HTML-коде. Эти значения указывают активные области изображения и содержат информацию о том, куда следует перейти браузеру после нажатия на одну из этих областей.

Активные области могут иметь форму прямоугольника, круга и многоугольника. Также позволительно комбинировать эти три варианта или определять область, расположенную вне активных областей карты-изображения.

Типы карт-изображений

Как уже было сказано, карты-изображения могут быть реализованы в двух вариантах — серверном и клиентском. Рассмотрим подробно каждый из них.

Серверный вариант

Серверный вариант реализации карты-изображения (Server-Side Imagemap) подразумевает, что документ, в котором прописаны координаты областей, находится на самом Web-сервере, который обязательно должен поддерживать выполнение CGI-сценария.

Процесс перехода по гиперссылкам на карте-изображении осуществляется следующим образом. При нажатии на одной из активных областей браузером передаются координаты на сервер, который обращается к специальному конфигурационному файлу. Результат обработки данных возвращается браузеру назад в виде адреса, соответствующего данной активной области, который загружается в окне обозревателя.

В случае если конфигурационный файл не содержит данных об искомой активной области, выводится сообщение об отсутствии данного документа. Для того чтобы браузер определил графический объект как карту-изображение, необходимо в теге `<IMG>`, описывающем конфигурацию карты-изображения, указать параметр ISMAP, а файл конфигурации сохранить в формате MAP.

Серверный вариант поддерживает два формата реализации карт-изображений — CERN и NCSA.

Первый разработан научным центром European Organization for Nuclear Research и предлагает следующую запись координат активных областей:

```
тип_области координаты адрес
```

Значения пар координат разделяются запятой и заключаются в круглые скобки, например:

```
rect (54, 127)(45, 344) http://www.site.ru/
```

В начале конфигурации стоит значение rect (прямоугольное выделение для задания активной области). Другими значениями могут быть: circle (круг). poly (многоугольник) и default (значение по умолчанию).

Формат NCSA разработан центром приложений для суперкомпьютеров National Center for Supercomputing Applications и предлагает несколько иной формат записи конфигурации карт-изображений:

```
тип_области адрес координаты
```

Координаты x, Y тоже разделяются запятыми, но в скобки не заключаются, например:

```
rect http://www.site.ru/ 54, 127  45, 344
```

Кроме типов областей, предложенных CERN, данный формат разрешает использование типа point (активизируется та ссылка после нажатия, которая обозначена ближе всего к точке соприкосновения).

Клиентский вариант

Клиентский вариант позволяет разместить все данные об активных областях карты-изображения в самом HTML-документе. В этом случае количество обращений к серверу сильно сокращается, а конфигурировать саму карту-изображение можно параллельно с изменением HTML-кода.

Размещать файл конфигурации на Web-сервере не нужно -- координаты активных областей указываются в самом документе, куда встроен графический объект для карты-изображения. При использовании клиентского варианта (Client-Side Imagemap) в тег <IMG> добавляется параметр USEMAP.

В связи с тем что клиентский вариант на сегодня распространен более широко и к тому же более доступен, подробно рассмотрим процесс конфигурации карты-изображения на примере Client-Side Imagemap.

Для определения конфигурации активных областей карты-изображения используется специальный тег-контейнер <MAP> с параметром NAME, который должен соответствовать свойству параметра USEMAP в теге <IMG>.

Необходимо отметить, что описание карты-изображения следует сразу же после указания тега рисунка <IMG>:

```
<IMG SRC="picture.gif" USEMAP="#mymap">
<MAP NAME="mymap">
Координаты активных областей...
</MAP>
```

Описание активных областей карты-изображения осуществляется с помощью тега <AREA>, не требующего за собой закрывающего тега.

Тег <AREA> может включать следующие параметры:

- SHAPE -- определяет форму активной области (значения такие же, как в формате CERN серверного варианта);
- COORDS -- осуществляет выбор конкретной активной области и содержит значения пар координат. Начало координат размещается в верхнем левом

углу графического изображения, которому соответствует начальное значение О, О;

- TARGET -- используется при работе с фреймами (рассмотрению фреймов посвящена *глава* 6);
- ALT — действие параметра аналогично использованию его в обычных гиперссылках на основе графических указателей (альтернативный текст);
- HREF -- наличие гиперссылки для данной области;
- NOHREF -- отсутствие гиперссылки для данной области.

Пример карты-изображения (клиентский вариант) приведен соответственно на рис. 5.6 и в листинге 5.3.

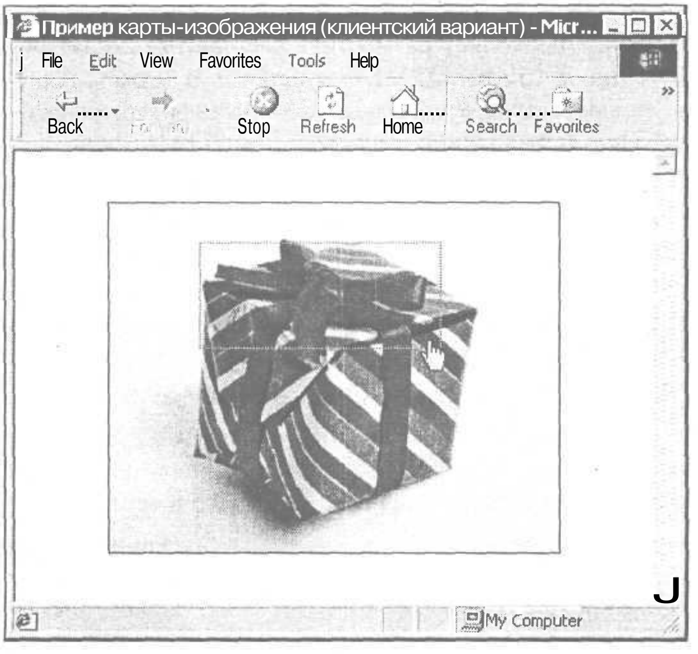


Рис. 5.6. Пример карты-изображения (клиентский вариант)

Листинг 5.3. Пример карты-изображения (клиентский вариант)

```
<HTML>
  <HEAD>
    <TITLE>Пример карты-изображения (клиентский вариант)</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
```

```
<IMG SRC="box.jpg" BORDER="1" WIDTH="300" HEIGHT="234" ALT=""
USEMAP="#gift">
<MAP NAME="gift">
<AREA SHAPE="rect" COORDS="60, 26, 222, 98" HREF="bantik.html"
ALT="Бантик">
<AREA SHAPE="rect" COORDS="63, 88, 135, 188" HREF="left.html" ALT="Левая
сторона">
<AREA SHAPE="rect" COORDS="151, 102, 225, 200" HREF="right.html"
ALT="Правая сторона">
</MAP>
</BODY>
</HTML>
```

На рис. 5.6 видна выделенная при нажатии активная область карты-изображения, ведущая по гиперссылке на документ "bantik.html".

В завершение следует добавить, что возможно совмещение серверного и клиентского вариантов карты-изображения. В этом случае в теге рисунка <IMG> необходимо указывать оба параметра — ISMAP и USEMAP. При этом следует помнить, что параметр USEMAP будет доминирующим.

Такой комбинированный вариант можно применять во избежание трудностей, связанных с отсутствием поддержки данной технологии некоторыми браузерами (хотя таких — считанные единицы). Если браузер не поддерживает Client-Side Imagemap, то параметр USEMAP проигнорируется, а реализован будет серверный вариант (при наличии конфигурационного файла на сервере и поддержке последним CGI).

Краткий обзор программ по работе с графикой

В этом разделе вашему вниманию представлен краткий обзор наиболее популярных и удобных программ для работы с графическими изображениями — графических редакторов, аниматоров, программ для просмотра и управления графическими изображениями, а также утилитами по созданию карт-изображений.

Графические редакторы

Adobe Photoshop 7

Мощный и многофункциональный редактор растровой графики от компании Adobe Systems (**http://www.adobe.com**) имеет полный спектр всех необходимых инструментов по обработке и ретушированию фотографических изображений, созданию потрясающей Web-графики и анимации.

В состав программного пакета входят две программы — Adobe Photoshop (ориентация на работу с фотоизображениями цифрового качества и разработку эффектной Web-графики) и Adobe ImageReady (оптимизация графики, подготовка изображений для размещения в Интернете, разработка анимации и карт-изображений).

Это достаточно сложный для освоения редактор, больше подходящий пользователю среднего уровня.

CorelDRAW Graphics Suite 11

Всемирно известный программный пакет по работе как с векторной, так и с растровой графикой от компании Corel Corporation (`http://www.corel.com`). CorelDRAW Graphics Suite предоставляет широкие возможности по разработке графических изображений, фоторедактированию и векторной анимации.

В состав пакета входят три программы — CorelDRAW (создание и редактирование векторной графики, полиграфия и предпечатная подготовка высококачественных иллюстраций), Corel PhotoPaint (работа с растровой графикой, фотоизображениями, создание и оптимизация Web-графики) и Corel R.A.V.E. (создание векторной анимации для HTML-документов).

Это также весьма сложный для обычного пользователя программный пакет.

Ulead PhotoImpact 8

Более простой и понятный для начинающего пользователя графический редактор от компании Ulead Systems (`http://www.ulead.com`). Условно его можно назвать облегченной версией Adobe Photoshop: программа обладает достаточно широкими функциональными возможностями, развитым инструментарием и интуитивно понятным рабочим интерфейсом.

Ulead PhotoImpact предназначен для создания и редактирования растровой графики, оптимизации и подготовки изображений с целью последующего размещения в Интернете. Содержит большой набор средств обработки и корректировки фотоизображений, библиотеку текстур, шаблонов и прочего полезного материала.

JASC PaintShop Pro 7

Компания JASC Software (`http://www.jasc.com`) также выпускает программный продукт, больше ориентированный на начинающего или среднего пользователя.

Растровый редактор JASC PaintShop Pro умеет работать с фотоизображениями, позволяет создавать и оптимизировать деловую, презентационную и Web-графику, содержит широкий набор эффектов, инструментов рисования и редактирования, включает программу Animation Shop для разработки анимационных роликов.

Аниматоры

Ulead GIF Animator 5

Мощная программа разработки анимации стандарта GIF от компании Ulead Systems (http://www.ulead.com). Содержит удобный рабочий интерфейс, широкий набор средств по оптимизации анимационного процесса. Включает библиотеку эффектов, а также возможность предпросмотра готовых роликов.

CoffeeCup GIF Animator 6.2

Очень простой и удобный аниматор от компании CoffeeCup Software (http://www.coffeecup.com). Программа имеет многочисленные награды, врученные на крупнейших зарубежных серверах программного обеспечения. Среди них такие, как ZDNet Editors' Pick, CNET Downloads Pick, Tucows и др.

Такое внимание со стороны общественности не случайно — CoffeeCup GIF Animator включает большой спектр возможностей для создания анимации. Поддержка технологии "перетаскивания" изображений (Drag and Drop), одновременный просмотр всего ролика и его отдельных кадров, экспортирование и импортирование видеофайлов формата AVI, генерация HTML-кода для вставки готовой анимации в электронный документ и многое другое.

Просмотрщики графических файлов

ACDSee 5.0

Одна из самых популярных во всем мире программ для просмотра и управления графическими изображениями от компании ACDSystems (http://www.acdsystems.com). ACDSee предоставляет пользователю широкий диапазон возможностей для просмотра, редактирования, ретуширования и печати фотографических изображений.

Поддержка большого количества растровых форматов, гибкие средства конвертирования, создание фотогалерей и слайдшоу, интеграция с периферийными устройствами (сканер, цифровая камера), развитые возможности предпросмотра и управления файлами практически любого формата.

XnView 1.5

Более простой и быстрый просмотрщик графических файлов от французского разработчика Пьера Гугеле (http://www.xnview.com).

Среди функциональных возможностей XnView особенно следует выделить следующие: свыше 300 различных форматов файлов для импортирования и около 40 для экспортирования, управление графическими файлами (копирование, перемещение и т. д.), ретуширование фотоизображений, применение

эффектов, полноэкранный режим, создание слайдшоу и галерей изображений, многоязыковая поддержка и многое другое.

Программы по созданию карт-изображений

Sausage Image Mapper 1.23

Простая утилита по созданию карт-изображений от компании Sausage Software (`http://www.sausage.com`). Выбор адреса гиперссылки, работа с областями выделения (создание, редактирование, удаление) и пр.

CoffeeCup Image Mapper

Еще один редактор карт-изображений от компании CoffeeCup Software (`http://www.coffeecup.com`). CoffeeCup Image Mapper имеет встроенный мастер карт-изображений, позволяет просматривать HTML-код и тестировать работу создаваемых карт-изображений, поддерживает основные области выделения, имеет расширенные средства работы с буфером обмена при редактировании активных областей и др.

Глава 6

HTML-документы на основе фреймов

Понятие фрейма

Фрейм (от англ. "frame" — рамка, каркас, кадр) представляет собой отдельное рабочее окно браузера, разделенное еще на несколько различных по параметрам и размеру фреймам. Совокупность таких окон принято называть фреймовой структурой.

Чаще всего такие структуры называют просто окнами (или же в соответствии с родным названием — фреймы). Фреймовая структура позволяет разбивать основную область на любое число составляющих подфреймов (подобластей), причем по необходимости определяя внутреннее поведение подфреймов.

Каждый фрейм — это не что иное, как отдельный HTML-документ, загружающийся браузером в предназначенном для него окне. Таким образом HTML-документ, созданный на фреймовой основе, является набором взаимосвязанных электронных документов, параметры и свойства которых определяются настройками всей фреймовой структуры.

Впервые возможность отображения фреймов была реализована компанией Netscape в одноименном браузере версии 2.0. Но ничего замечательного первоначальный вариант поддержки подобной структуры не предлагал своим пользователям. Однако уже с выходом версии 3.0 в браузере Netscape появился ряд нововведений, расширивших диапазон параметров основных тегов описания фреймов.

Компания Microsoft тоже не тратила время впустую и, решив потеснить Netscape на перспективном рынке браузеров, выпустила летом 1996 года свой новый продукт — Microsoft Internet Explorer 3.0, который поддерживал практически все основные расширения Netscape и к тому же предполагал реализацию так называемых плавающих фреймов.

Далее и та и другая компания старались покорить пользователя посредством расширения возможностей фреймовых структур, которые поддерживались

только собственными браузерами. К сожалению, и на сегодняшний день не всегда можно быть уверенным в том, что электронный документ, созданный с применением многооконной (фреймовой) структуры, одинаково корректно отобразится в браузерах обеих компаний.

Тем не менее, в настоящее время фреймы получили достаточное распространение и представляют собой еще один вариант системы навигации интернет-ресурса.

Области применения фреймов

Диапазон применения фреймов не настолько широк как, например, у карт-изображений или меню навигации в виде обыкновенных текстовых гиперссылок. Чаще всего прибегают к фреймовой структуре в следующих случаях:

- создание неподвижной или прокручиваемой навигационной панели управления;
- одновременное отображение информации в нескольких местах;
- постоянное визуальное присутствие определенного текстового, графического или иного объекта;
- разработка Web-интерфейса для онлайн-игр.

Рассмотрим подробнее каждую из названных областей применения.

Панель навигации

Независимо от типа интернет-ресурса (корпоративный сервер, персональная страница, общественная организация и т. д.) в последнее время стало весьма популярным применение фреймов для создания привлекательных и удобных навигационных панелей (меню). Эти панели могут быть статичными (т. е. не зависящими от действий посетителя) и динамическими, когда какое-либо действие со стороны пользователя (нажатие комбинации клавиш или на кнопку мыши и пр.) приводит к различным по виду и масштабу изменениям, начиная с замены внешнего вида навигационных кнопок (эффект RollOver/MouseOver) и заканчивая исполнением сложных программных сценариев.

Такая реализация фреймов удобна прежде всего тем, что она позволяет посетителю перемещаться по сайту и независимо от его месторасположения (двадцать страниц он пролистал или всего одну) постоянно иметь перед глазами панель управления с перечнем всех или основных разделов сайта.

Реализовать такую панель управления можно разными способами:

- в главном фрейме обозначить только основные разделы Web-сайта, а подразделы открывать в другом окне;

- в главном фрейме разместить ссылки на документы подразделов, тогда при нажатии на ссылку конкретного подраздела главный фрейм перезагрузится и откроется расширенный вариант меню;
- в главном меню расположить динамическое меню в виде раскрывающегося списка, по нажатию на пункты которого в другом фрейме будет открываться нужный документ.

На рис. 6.1 представлен пример меню навигации, составленного на основе фреймовой структуры.

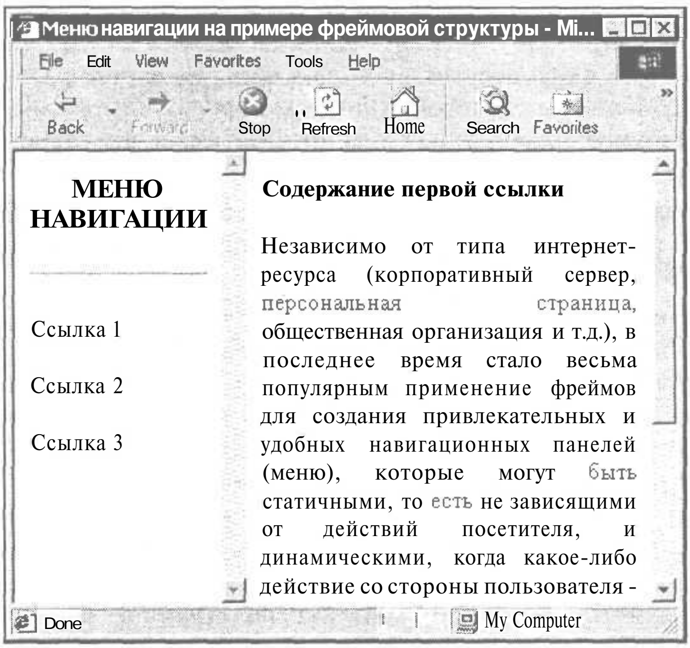


Рис. 6.1. Меню навигации на примере фреймовой структуры

Одновременное отображение информации

Иногда бывает необходимо иметь перед глазами одновременно несколько текстовых или иных информационных блоков, расположенных в разных окнах. Например, сравнительные характеристики котировок акций (покупка-продажа), структура книги (главы, разделы, подразделы и соответствующее им содержание), перечень товаров и их разнообразные характеристики и др.

Согласитесь, что было бы весьма неудобно и непрактично размещать раздельно документы с данными о покупке и документы с информацией о продаже акций, публиковать электронную версию книги без оглавления, растягивать знакомство с товаром на продолжительное время, включающее

переход с одной страницы (название и фирма-производитель) к другой (изображение товара), тем более что подобный переход может состоять еще из нескольких стадий. Все это чревато ослаблением внимания посетителя как потенциального покупателя и потерей интереса к товару.

Постоянное визуальное присутствие объекта

Иногда разработчики HTML-документов ставят перед собой задачу размещения конкретного объекта на странице таким образом, чтобы он был хорошо виден посетителям независимо от их действий (конечно, в пределах данного Web-сайта).

То есть что бы ни делал посетитель — переходил из одного раздела в другой, оставлял сообщение в гостевой книге, копировал интересный материал, просматривал документы и т. д., — данный объект в визуальном плане всегда должен быть доступен.

В качестве таких объектов могут выступать фирменные логотипы, эмблемы, фотографии, рекламные баннеры, навигационные меню, надписи, рубрикаторы и многое другое.

Что касается навигационных меню и баннеров, то возможно создание таковых при помощи технологий DHTML и JavaScript, которые позволяют разработчику таким образом определить характеристики меню, чтобы оно всегда было видно пользователю. Например, пользователь немного прокрутил окно вниз — меню автоматически тоже опускается чуть-чуть вниз, документ прокручивается вверх — движение меню полностью повторяет поведение страницы.

Однако не следует забывать о том, что многих такое дрейфующее положение рекламы и меню сильно раздражает, а порой мешает рассмотреть некоторые элементы на самой Web-странице. Аналогичным вариантом реализации эффекта постоянного присутствия объекта на странице будет применение многооконной (фреймовой) структуры.

Web-интерфейсдляонлайн-игр

Наконец, фреймы применяются не только на познавательно-информационных сайтах и коммерческих серверах. С их помощью можно создавать Web-интерфейсы для самых разнообразных игр в режиме онлайн.

Преимущества и недостатки фреймов

Само собой разумеется, что фреймовая структура, как и любое другое технологическое решение, имеет своих противников и сторонников. Кому-то фреймы не нравятся исключительно из визуальных соображений (нажимаешь в одном месте, а все меняется уже в другом), другим пользователям фреймы не нравятся по ряду технических причин.

Рассмотрим основные преимущества и недостатки применения фреймовых структур. Начнем, как водится, с положительных сторон:

П фреймы позволяют экономить на объеме пересылаемых пользователю файлов, поскольку после активизации ссылки изменяется только один фрейм;

❒ фреймы заметно облегчают навигацию по электронным документам благодаря возможности перехода по другим ссылкам в пределах интернет-ресурса; возможность работы сразу с несколькими информационными блоками в пределах одного окна позволяет экономить время;

П использование правил описания фреймовых структур позволяет разработчику HTML-документов как угодно варьировать размеры полей фреймов, что дает более широкий спектр возможностей пространственного размещения объектов.

А теперь — несколько недостатков фреймов:

П некоторые поисковые механизмы не в состоянии индексировать документы с фреймовой структурой или делают это не совсем корректно, что приводит к индексированию не родительского фрейма, как требуется, а одного из его составляющих;

П компоновка страниц с фреймами часто выглядит неуклюже и грубо, многие не понимают, как пользоваться его возможностями;

П на Web-сайтах, использующих фреймовые структуры, при перемещении назад/вперед в адресной строке ничего не меняется — там постоянно указан начальный адрес сайта;

П некоторые малоизвестные браузеры (а также ранние версии популярных) при попытке перейти назад к предыдущему документу, который только что просматривался, возвращаются в начало Web-сайта. То же самое происходит, если попробовать обновить страницу с фреймовой структурой.

Правила описания фреймов

Для описания фреймовых структур в HTML предусмотрен ряд специальных тегов и параметров, которые сейчас и рассмотрим.

Тег *<FRAMESET>*

Тег `<FRAMESET>` является основным тегом-контейнером фреймовой структуры HTML-документа (закрывающий тег обязателен). Особенностью любого электронного документа с фреймами является то, что в HTML-коде отсутствует тег-контейнер <BODY>, тег `<FRAMESET>` вставляется сразу же после раздела HEAD.

Структурный HTML-документ (тот, который определяет структуру фреймов) не может содержать ни тегов форматирования, ни каких-либо HTML-элементов.

Тег <FRAMESET> может содержать два основных параметра — ROWS и COLS.

Параметры ROWS и COLS определяют количество горизонтальных (ROWS) и/или вертикальных (COLS) фреймов. Формат записи значения может быть в пикселах, процентах или относительных единицах, причем число значений соответствует количеству фреймов.

Значения записываются через запятую, например:

```
<FRAMESET ROWS="30%, 70%">
```

В данном примере мы имеем два горизонтальных фрейма, один из которых (верхний) занимает 30% рабочей области окна браузера, а второй (нижний) — 70% (общая сумма всегда должна составлять 100%).

Запись значений в пикселах не очень удобна ввиду того, что разработчик фреймовой структуры (да и вообще всего HTML-документа) не знает, на каком мониторе и при каком экранном разрешении будет просматриваться его документ. В этом смысле оптимально подходит указание значений в процентном соотношении — при изменении размеров окна браузера размеры фреймов будут пропорционально меняться.

Формат записи значений в относительных единицах выглядит так:

```
<FRAMESET COLS="*, 2*, 3*">
```

Символ "звездочки" (*) является одной частью целого числа и осуществляет пропорциональное деление окна браузера на указанное количество фреймов (в нашем случае это три вертикальных фрейма). Чтобы определить дробное число, соответствующее размеру конкретного фрейма, нужно сложить все цифры, стоящие перед символом *, и подставить получившуюся сумму в качестве знаменателя от целого числа. Таким образом: 1 + 2 + 3 = 6, т. е. первый фрейм занимает 1/6 окна, второй — 2/6 (или 1/3) окна, а третий — 3/6 (или 1/2) окна браузера (рис. 6.2). Отсутствие цифры перед символом "звездочки" подразумевает значение 1.

Язык разметки HTML позволяет для параметров ROWS и COLS задавать смешанные значения, например:

```
<FRAMESET ROWS="50, 50%, *, 3*">
```

Такая структура включает в себя четыре горизонтальных фрейма: первый строго зафиксирован (50 пикселов), второй занимает 50% от всего размера окна браузера, а два последних фрейма поделят оставшееся пространство в соотношении 1/4 и 3/4.

Обязательного порядка для записи смешанных значений не существует, однако рекомендуется в первую очередь указывать фиксированные значения (пикселы), затем проценты, а уже потом относительные единицы.

Также в тег <FRAMESET> возможно включать оба параметра — ROWS и COLS (рис. 6.3):

```
<FRAMESET COLS="40%, 60%" ROWS="*, 2*">
```

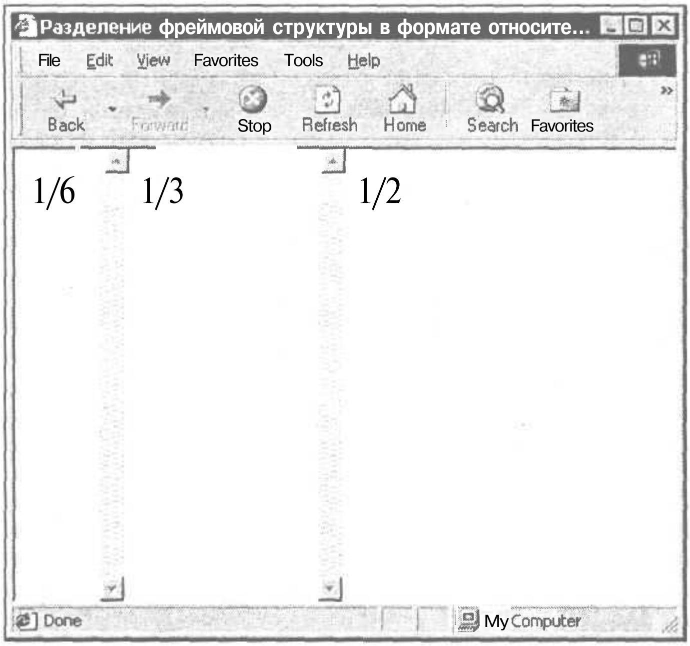


Рис. 6.2. Разделение фреймовой структуры в формате относительных единиц

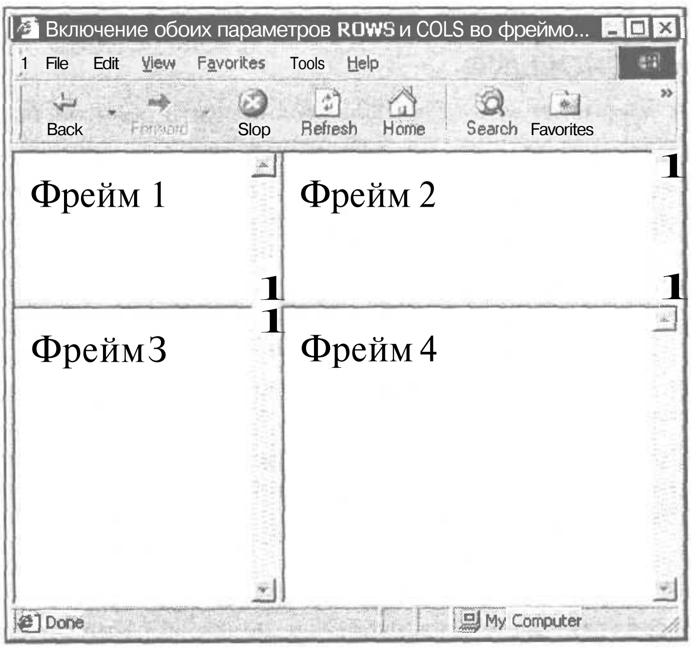


Рис. 6.3. Включение обоих параметров ROWS и COLS во фреймовую структуру

Тег <FRAME>

Тег <FRAME> описывает свойства отдельного фрейма, входящего в состав фреймовой структуры. Соответственно, сколько имеется фреймов, столько должно быть и тегов <FRAME>, заключенных внутри тега фреймовой структуры <FRAMESET>. Тег <FRAME> не требует закрывающего тега и может содержать параметры, которые мы сейчас и рассмотрим.

Параметр *SRC*

Параметр SRC указывает путь к HTML-документу, занимающему текущий фрейм:

```
<FRAME SRC="frames/menu.html">
```

В данном случае браузером во фрейм будет загружен документ «menu.html". В качестве источника фрейма может быть указан любой допустимый адрес.

Также следует помнить, что свойства документа, являющегося составным компонентом фреймовой структуры, определяются в HTML-коде этого документа, а не в пределах конструкций <FRAMESET> или <FRAME>.

Параметр *NAME*

Данный параметр присваивает текущему фрейму внутреннее имя, которое используется для взаимодействия между различными фреймами (подробнее изложено в *разделе "Взаимодействие между фреймами" этой главы).*

Параметр *SCROLLING*

С помощью параметра SCROLLING можно управлять полосой прокрутки в отдельном фрейме. Существуют три возможных значения данного параметра:

- YES - полоса прокрутки появляется в любом случае. Такое значение не рекомендуется присваивать, поскольку постоянное присутствие этой полосы может негативно сказаться на дизайне всего электронного документа;
- NO — запрет полосы прокрутки. Прокрутка не появится даже в случае, если содержание документа превышает размер видимого пространства окна браузера. В итоге оставшаяся часть документа будет недоступна пользователю;
- AUTO -- автоматическое появление полосы прокрутки по мере необходимости. Это рекомендуемое значение, оно регулирует прокрутку в зависимости от объема информации в текущем фрейме.

Параметр *NORESIZE*

Параметр NORESIZE запрещает ручное изменение пользователем размеров текущего фрейма. Такое изменение возможно при наведении курсора мыши

на границу фрейма, однако в ряде случаев разработчику может потребоваться запретить изменение размеров окон фреймовой структуры.

Параметр NORESIZE не требует указания каких-либо значений. Также следует помнить о том, что при наличии данного параметра изменить размеры соседних фреймов тоже будет нельзя.

Параметры MARGINWIDTHи MARGINHEIGHT

Параметры MARGINWIDTH и MARGINHEIGHT предназначены для определения горизонтальных и вертикальных отступов внутри фрейма. Значение указывается в пикселах и является равнозначным для обеих сторон, например:

```
<FRAME MARGINWIDTH="5" MARGINHEIGHT="3">
```

В данном случае горизонтальный отступ будет равен 5 пикселам, а вертикальный равен 3 пикселам.

При указании этих двух параметров необходимо помнить о том, что такие же отступы могут быть определены не только в файле фреймовой структуры, но и непосредственно в HTML-коде документа выбранного фрейма (параметры LEFTMARGIN, RIGHTMARGIN, TOPMARGIN, BOTTOMMARGIN, MARGINWIDTH И MARGINHEIGHT в теге <BODY>).

Тег <NOFRAMES>

Возможность работы с фреймами не была предусмотрена спецификацией HTML вплоть до версии 3.2. И это несмотря на то, что многие браузеры данную технологию все же поддерживали. С принятием HTML 4.0 была введена поддержка фреймовых структур, а Консорциум W3C стал рекомендовать использование тега-контейнера <NOFRAMES> (закрывающий тег обязателен), предоставляющего информацию пользователям браузеров, не поддерживающих фреймовую структуру.

```
<NOFRAMES>
Этот документ содержит фреймы.
</NOFRAMES>
```

Текст, заключенный между двумя тегами <NOFRAMES>, будет показан владельцам таких браузеров, а современные браузеры просто проигнорируют эту информацию.

Дополнительные параметры фреймовых структур

К дополнительным параметрам фреймовых структур можно отнести параметры, которые поддерживаются не всеми браузерами.

Это параметры BORDER (толщина границы между фреймами), FRAMEBORDER (наличие границы между фреймами), BORDERCOLOR (цвет границы фрейма) и FRAMESPACING (отступ между соседними фреймами).

В завершение разговора о правилах построения фреймовых структур рассмотрим итоговый пример (рис. 6.4, листинг 6.1).

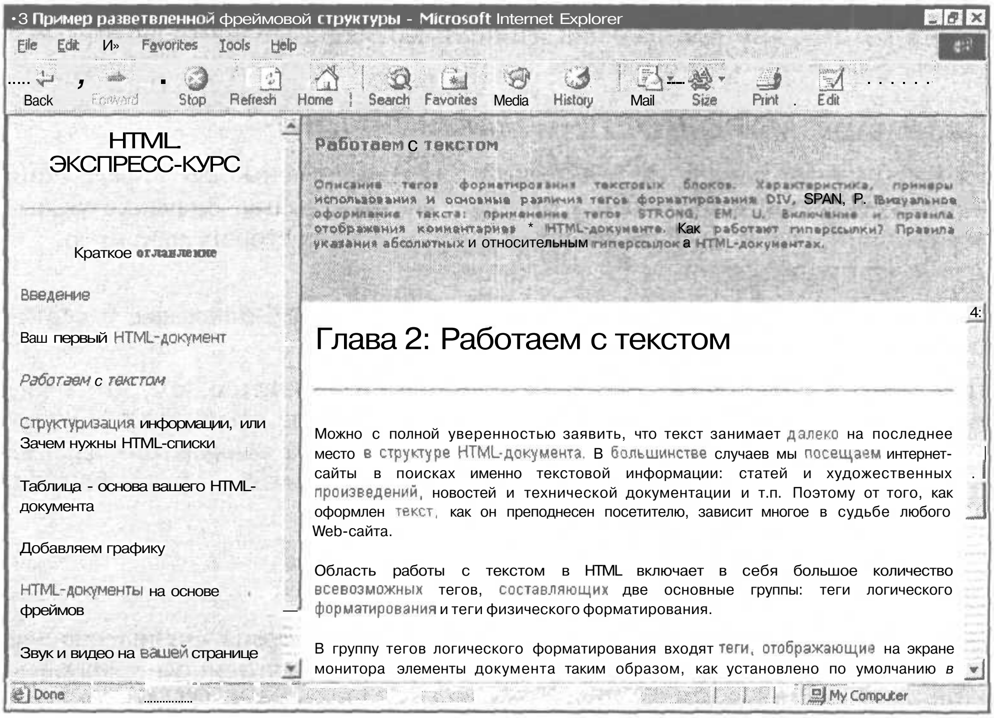


Рис. 6.4. Пример разветвленной фреймовой структуры

Листинг 6.1. Пример разветвленной фреймовой структуры

```
<HTML>
  <HEAD>
    <TITLE>Пример разветвленной фреймовой структуры</TITLE>
  </HEAD>
<FRAMESET COLS="30%, 70%" FRAMEBORDER="0" FRAMESPACING="0" BORDER="0">
<FRAME NAME="left" SRC="left.html" SCROLLING="yes" NORESIZE
MARGINWIDTH="10" MARGINHEIGHT="10">
<FRAMESET ROWS="*, 2*" FRAMEBORDER="0" FRAMESPACING="0" BORDER="0">
<FRAME NAME="top" SRC="top.html">
<FRAME NAME="bottom" SRC="bottom.html">
</FRAMESET>
<NOFRAMES>
```

```
Ваш браузер не поддерживает фреймы.
</NOFRAMES>
</HTML>
```

Пример описывает фреймовую структуру, состоящую из двух вертикальных и двух горизонтальных фреймов.

Конструкция

```
<FRAMESET COLS="30%, 70%" FRAMEBORDER="0" FRAMESPACING="0" BORDER="0">
```

показывает, что окно браузера поделено на две части: 30% и 70%, соответственно. Следующий за ней тег <FRAME> задает свойства левого фрейма.

Правый фрейм разбит еще на два фрейма, на этот раз горизонтальных, для чего дается отдельная конструкция:

```
<FRAMESET ROWS="*, 2*" FRAMEBORDER="0" FRAMESPACING="0" BORDER="0">
```

После этого посредством тега <FRAME> дается описание двух упомянутых горизонтальных фреймов. Завершает код файла фреймовой структуры тег-контейнер <NOFRAMES>, предназначенный для браузеров, не поддерживающих фреймы.

Для закрепления пройденного материала основные теги фреймовых структур и их параметры приведены в табл. 6.1.

Таблица 6.1. Основные теги и параметры фреймовых структур

Тег	Параметры	Функция
FRAMESET	COLS ROWS	Основной тег фреймовой структуры
FRAME	SRC NAME SCROLLING NORESIZE MARGINWIDTH MAGINHEIGHT	Тег описания свойств отдельного фрейма
NOFRAMES		Тег информации для браузеров без поддержки фреймов

Взаимодействие между фреймами

Работа с фреймовыми структурами имеет свои специфические особенности, которые следует знать, если имеется желание использовать данное средство навигации для HTML-документов.

Взаимодействие между отдельными фреймами происходит посредством загрузки документа по соответствующей ссылке в установленное окно. Реализуется это действие с помощью параметра TARGET тега <A>.

По умолчанию вызванный по гиперссылке документ будет загружаться в текущий фрейм. Но чаще всего нужно сделать так, чтобы страница появлялась в соседнем окне. Существуют специальные зарезервированные имена действий, по которым происходит загрузка документов на сайтах с фреймовой структурой: "_blank", "_self", "_parent" И "_top".

Действие

```
<A HREF="file1.html" TARGET="_blank">
```

осуществляет загрузку документа в новое окно без имени, определяемого параметром NAME тега <FRAME>, поэтому этот вариант исключает изменение содержимого созданного окна.

Действие

```
<A HREF="file2.html" TARGET="_self">
```

открывает документ в текущем окне.

Действие

```
<A HREF="file3.html" TARGET="_parent">
```

производит загрузку документа в область, занимаемую родительским фреймом текущего фрейма.

Действие

```
<A HREF="file4.html" TARGET="_top">
```

вызывает загрузку документа в полном окне.

В данном случае не был задействован параметр NAME в теге описания текущего фрейма <FRAME>.

Рассмотрим несколько примеров по взаимодействию между фреймами и открытию отдельных окон браузера с помощью параметра TARGET.

Создадим файл со следующей фреймовой структурой:

```
<FRAMESET ROWS="2*, *">
<FRAME NAME="menu" SRC="menu.html" NORESIZE FRAMEBORDER="1">
<FRAMESET COLS="50%, 50%">
<FRAME NAME="left" SRC="left.html">
<FRAME NAME="right" SRC="right.html">
</FRAMESET>
```

Верхний фрейм будет содержать перечень гиперссылок, а два нижних фрейма предназначены для открытия в них содержимого конкретной ссылки.

Документ "menu.html" (рис. 6.5, листинг 6.2) содержит шесть гиперссылок на один и тот же файл — "text.html".

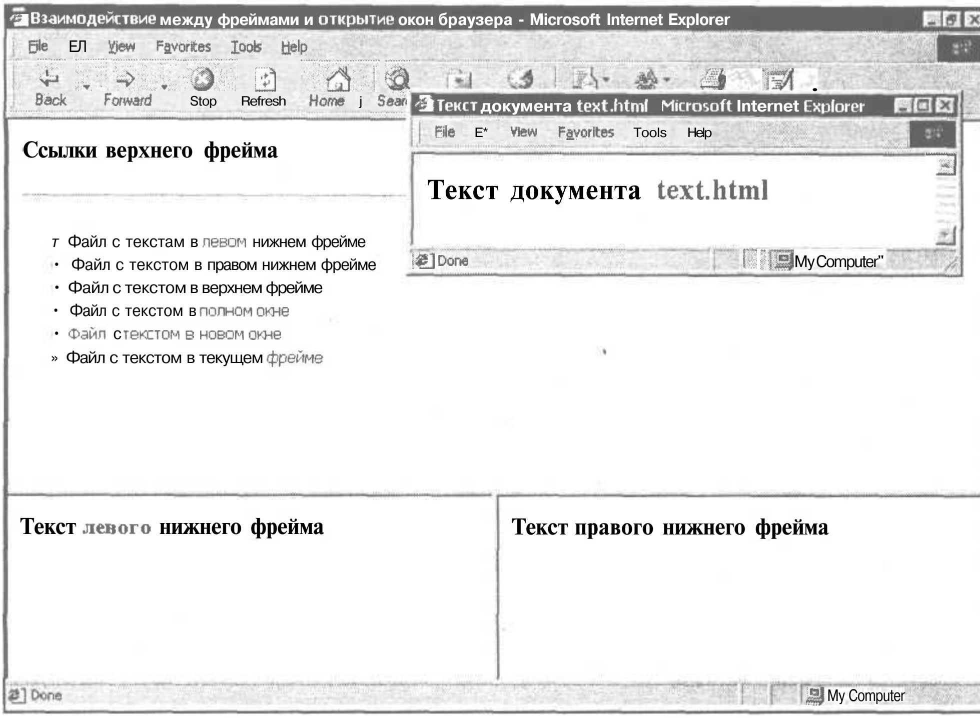


Рис. 6.5. Взаимодействие между фреймами и открытие окон браузера

Листинг 6.2. Файл верхнего фрейма "menu.html"

```
<HTML>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#00FF00"
VLINK="blue">
<H3>Ссылки верхнего фрейма</H3>
<HR>
<FONT FACE="Tahoma" SIZE="2">
<UL TYPE="square">
<LIXA HREF="text.html" TARGET="left">Файл с текстом в левом нижнем фрей-
ме</A>
<LIXA HREF="text.html" TARGET="right">Файл с текстом в правом нижнем
фрейме</A>
<LIXA HREF="text.html" TARGET="menu">Файл с текстом в верхнем фрейме</A>
<LIXA HREF="text.html" TARGET="_top">Файл с текстом в полном окне</A>
<LI><A HREF="text.html" TARGET="_blank">Файл с текстом в новом окне</A>
<LIXA HREF="text.html" TARGET="_self">Файл с текстом в текущем фрейме</A>
```

```
</UL>
</FONT>
</BODY>
</HTML>
```

Первая ссылка откроется в левом нижнем фрейме по причине указания конструкции `TARGET="left"` ("left" в данном случае — это внутреннее имя данного фрейма).

Вторая ссылка откроется в правом нижнем фрейме (указано внутреннее имя фрейма `"right"`).

Третья ссылка будет открыта в этом же окне, т. к. "menu" — это имя текущего фрейма, из которого открываются гиперссылки.

Четвертая ссылка откроется в полном окне браузера (конструкция `"_top"`).

Пятая ссылка будет открыта в отдельном новом окне поверх фреймовой структуры (действие `"_blank"`).

Наконец, последняя ссылка откроется в текущем фрейме (аналогично действию `TARGET="menu"`).

Плавающие фреймы

Плавающие фреймы являются стандартом, который поддерживается только браузером Internet Explorer. Netscape подобные фреймы просто-напросто не отображает.

Суть плавающих фреймов заключается в возможности встраивать обыкновенные фреймы (с указанием источника), являющиеся в сущности HTML-документами, в другие электронные документы.

Плавающие фреймы описываются тегом-контейнером `<IFRAME>` и могут иметь параметры, присущие обыкновенным фреймам, а также параметры, аналогичные описанию графических изображений (а именно: ALIGN, WIDTH, HEIGHT, HSPACE И VSPACE) (листинг 6.3, рис. 6.6.).

Листинг 6.3. Пример плавающих фреймов

```
<HTML>
<HEAD>
    <TITLE>Пример плавающих фреймов</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#00FF00"
VLINK="blue">
<IFRAME SRC="iframe.html" NAME="iframe" WIDTH="150" HEIGHT="250"
HSPACE="5" VSPACE="5" SCROLLING="auto" ALIGN="right">
```

```
Ваш браузер не отображает плавающие фреймы.
</IFRAME>
<P ALIGN="justify">
Плавающие фреймы являются стандартом, который поддерживается только брау-
зером Internet Explorer. Netscape подобные фреймы просто-напросто не ото-
бражает.
<BRXBR>
Суть плавающих фреймов заключается в возможности встраивать обыкновенные
фреймы (с указанием источника), являющиеся в сущности HTML-документами, в
другие электронные документы.
</P>
</BODY>
</HTML>
```

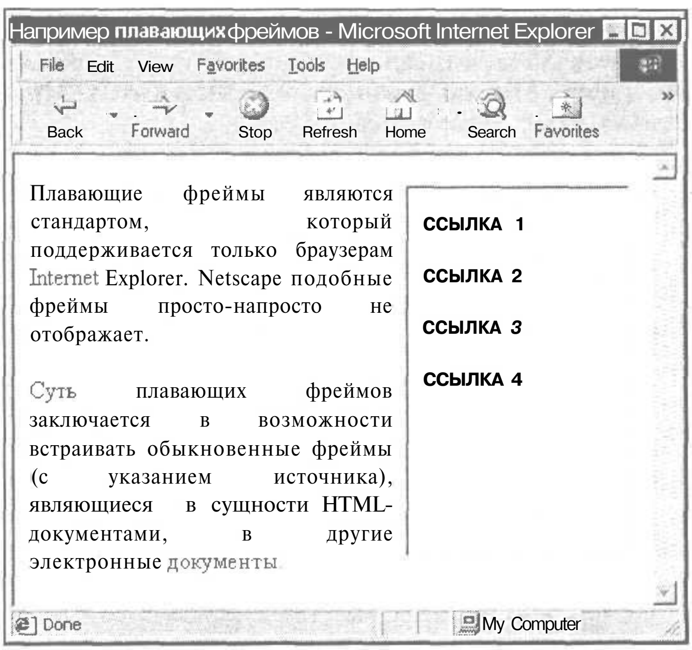


Рис. 6.6. Пример плавающих фреймов

Браузеры, которые не поддерживают плавающие фреймы, в указанном месте документа выведут сообщение "Ваш браузер не отображает плавающие фреймы".

Глава 7

Звук на вашей странице

В нашей повседневной жизни звук играет далеко не последнюю роль. Мы слушаем радио и музыку, записанную на CD, ежедневно слышим звук из динамиков наших телевизоров, устанавливаем затейливые мелодии для будильников, даже поздравительные открытки — и те стали музыкальными.

Развитие компьютерных технологий, в частности Интернета, на сегодняшний день позволяет применять в электронных документах компоненты звука. Существует несколько возможных вариантов воспроизведения звуковых файлов в Интернете. Мы же в рамках знакомства с HTML остановимся именно на возможностях этого языка разметки.

Звуковые форматы

Существует огромное количество звуковых форматов — распространенных и не особо встречающихся, применяемых в различных операционных системах и для различных нужд.

Нас интересуют те форматы, файлы которых позволительно встраивать в HTML-документы и воспроизводить непосредственно в браузере пользователя. К ним можно отнести:

- WAV формат звуковых файлов операционной системы Microsoft Windows;
- MIDI — Musical Instrument Digital Interface (формат электронных инструментов);
- AU — формат Sun/NeXT;
- AIFF — формат платформы Macintosh;
- RealAudio — формат передачи звуковых файлов в режиме реального времени.

Также в последнее время большую популярность получил формат MP3 (формат сжатия аудиосигнала с потерей качества, лежащей за пределами возможностей человеческого слуха). Однако с помощью стандартных средств HTML файлы MP3 встроить в структуру электронного документа нельзя.

Встраивание звуковых файлов в HTML-документ

Для встраивания практически любых звуковых файлов в HTML-документ можно использовать два пути:

- создание гиперссылки на музыкальный файл и размещение ее на странице;
- применение специального тега вставки музыкальных файлов <EMBED>.

Создание гиперссылки на музыкальный файл

Правила записи гиперссылки на файлы музыкальных форматов абсолютно идентичны записи ссылок на HTML-документы или графику:

```
<A HREF="music.wav">Музыкальный файл (30 Кб)</A>
```

При нажатии на такую ссылку на компьютере пользователя запускается установленный по умолчанию проигрыватель звуковых файлов.

Следует обратить особое внимание на указание размера музыкального файла, стоящее в ссылке сразу после названия. Дело в том, что некоторые звуковые данные (песни, фрагменты мелодий, фонограммы и пр.) могут иметь весьма большой размер. И прежде чем нажать на подобную ссылку, пользователь, учитывая скорость соединения с Интернетом и технические возможности своего компьютера, должен иметь хотя бы примерное представление о том, сколько времени займет открытие файла.

Применение тега *<EMBED>*

Тег <EMBED> имеет ряд параметров, которые позволяют управлять воспроизведением музыкального файла непосредственно в окне браузера. Пример встраивания звукового файла с помощью тега <EMBED> приведен соответственно в листинге 7.1 и на рис. 7.1.

Листинг 7.1. Пример встраивания звукового файла с помощью тега <EMBED>

```
<HTML>
  <HEAD>
    <TITLE>Пример встраивания звукового файла</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#00FF00"
VLINK="blue">
<H2 ALIGN="center">Пример встраивания звукового файла</H2>
```

```
<CENTER>
<EMBED SRC="melody.wav" WIDTH="287" HEIGHT="213" AUTOSTART="true">
</CENTER>
</BODY>
</HTML>
```

В этом случае браузер отобразит встроенную панель воспроизведения указанного музыкального файла ("melody.wav").



Рис. 7.1. Пример встраивания звукового файла с помощью тега <EMBED>

Параметры тега <EMBED> приведены в табл. 7.1.

Таблица 7.1. Параметры тега <EMBED>

Параметр	Описание
SRC	Указание пути к воспроизводимому звуковому файлу форматов AU, AIFF, WAV и MIDI. Является обязательным параметром
WIDTH	Ширина панели воспроизведения (обязательный параметр)
HEIGHT	Высота панели воспроизведения (обязательный параметр)
AUTOSTART	Значение по умолчанию — FALSE. Для автоматического начала воспроизведения используется значение TRUE

Таблица 7.1 (окончание)

Параметр	Описание
AUTOLOAD	Значение по умолчанию — TRUE. Для запрета автоматической загрузки файла используется значение FALSE
ALIGN	Тип выравнивания панели воспроизведения относительно текста на странице. Возможные значения: LEFT, RIGHT, CENTER, TOP и BASELINE. Значение по умолчанию — BASELINE
VOLUME	Установка громкости воспроизведения (определяется в процентах)
STARTTIME	Время начала воспроизведения в минутах и секундах (формат записи: "mm: ss"; по умолчанию — 00.00)
ENDTIME	Время окончания воспроизведения в минутах и секундах (формат записи: "mm: ss"; по умолчанию — конец звукового файла)
CONTROLS	Указание элементов управления на панели воспроизведения (значения CONSOLE, SMALLCONSOLE, PLAYBUTTON, PAUSEBUTTON, STOPBUTTON, VOLUMELEVER). Значение по умолчанию — CONSOLE
CONSOLE	Отображение полного набора элементов управления на панели воспроизведения
SMALLCONSOLE	Компактный набор элементов управления на панели воспроизведения
PLAYBUTTON	Отображение только кнопки воспроизведения
PAUSEBUTTON	Отображение только кнопки паузы
STOPBUTTON	Отображение только кнопки остановки воспроизведения (звуковой файл при этом будет выгружен)
VOLUMELEVER	Отображение только регулятора громкости

Особое внимание при встраивании музыкальных файлов в HTML-документы следует обратить на совместимость перечисленных параметров тега <EMBED> с различными браузерами.

Дело в том, что браузер Internet Explorer использует для воспроизведения встроенных звуковых файлов модуль ActiveMovie (проигрыватель Windows Media), а Netscape — LiveAudio. Каждый из этих модулей имеет ряд своих преимуществ, однако результат встраивания музыкального файла визуально может сильно отличаться при просмотре в этих двух браузерах.

RealAudio

Формат RealAudio был разработан компанией RealNetworks в период развития аудио- и видеоконференций в режиме реального времени. Новый формат положил начало возможности передавать звуковые файлы в Интернете в реальном времени.

Технология RealAudio состоит из трех программных компонентов:

- кодировщик — конвертирование звуковых файлов в формат RealAudio;
- сервер — передача файлов формата RealAudio по сети;
- проигрыватель — воспроизведение музыкальных файлов в формате RealAudio.

Для прослушивания RealAudio-файлов достаточно иметь установленный проигрыватель. Для распространения собственных файлов придется обзавестись и программой-кодировщиком, и сервером.

Для встраивания звукового файла в формате RealAudio на HTML-страницу используется также два варианта - посредством ссылки на источник RealAudio-данных и с помощью уже известного нам тега <EMBED>.

Ссылка на файл RealAudio выглядит так:

```
pnm://audio.real.com/melody.ra
```

где:

pnm: // — это указание сервера RealAudio;

audio.real.com — адрес сервера RealAudio;

melody.ra — название звукового файла RealAudio (расширение ra).

Совокупность ссылок такого вида обычно размешают в обыкновенном текстовом файле, содержащем информацию о звуковых данных RealAudio, которому присваивают расширение ram (RealAudio Metafile).

Таким образом позволительно указание обоих расширений (и ra, и ram), однако вариант с расширением ram предоставляет дополнительные возможности по определению точки проигрывания файла и даты его воспроизведения.

```
pnm://audio.real.com/melody.ra$0:30
```

В этом примере файл формата RealAudio будет воспроизводиться с тридцатой секунды (запись типа $мин:сек). Дата начала проигрывания определяется конструкцией $dd:hh:mm:ss:t (день:час:минута:секунда:десятая доля секунды).

Также можно вставить в HTML-документ обыкновенную гиперссылку, при нажатии на которую на компьютере пользователя запустится проигрыватель файлов RealAudio (рис. 7.2):

```
<A HREF="audio/melody.ra">Файл в формате RealAudio</A>
```

При встраивании файлов RealAudio с помощью тега <EMBED> необходимо помнить, что расширение файла должно быть не ram, а rpm (для активизации подключаемого к браузеру модуля воспроизведения звуковых файлов RealAudio). Пример встраивания файлов RealAudio в HTML-документ приведен соответственно на рис. 7.3 и в листинге 7.2.

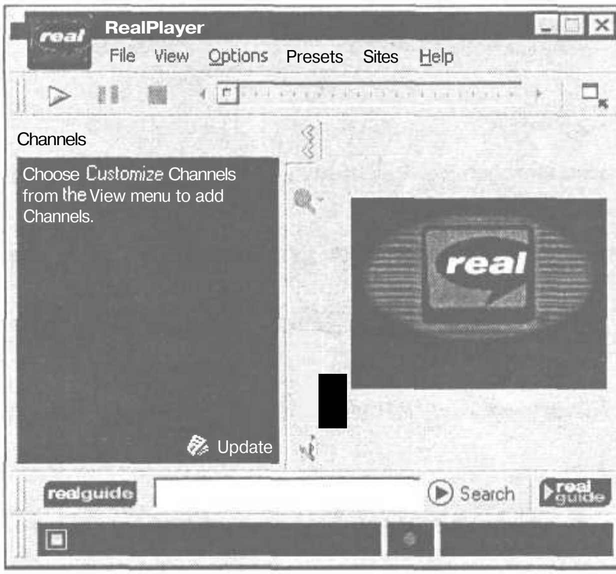


Рис. 7.2. Окно проигрывателя звуковых файлов RealPlayer



Рис. 7.3. Встраивание файлов RealAudio в HTML-документ

Листинг 7.2. Встраивание файлов RealAudio в HTML-документ

```
<HTML>
  <HEAD>
    <TITLE>Встраивание файлов RealAudio в HTML-документ</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#FF0000" ALINK="#00FF00"
VLINK="blue">
<H2 ALIGN="center">Встраивание файлов RealAudio</H2>
<CENTER>
<EMBED SRC="audio/melody.rpm" WIDTH="300" HEIGHT="128" AUTOSTART="true"
CONTROLS="All">
</CENTER>
</BODY>
</HTML>
```

Параметры тега `<EMBED>` для воспроизведения звуковых файлов в формате RealAudio перечислены в табл. 7.2.

Таблица 7.2. *Параметры тега `<EMBED>` (RealAudio)*

Параметр	Описание
SRC	Указание пути к воспроизводимому звуковому файлу формата RealAudio. Является обязательным параметром
WIDTH	Ширина панели воспроизведения (обязательный параметр)
HEIGHT	Высота панели воспроизведения (обязательный параметр)
CONTROLS	Набор элементов управления на панели воспроизведения (полный перечень возможных значений приведен в табл. 7.3)
AUTOSTART	При значении TRUE проигрывание файла начинается автоматически (порядок загрузки и последовательность воспроизведения нескольких файлов в формате RealAudio зависит от настроек браузера и программы-сервера)
CONSOLE	Создание нескольких панелей воспроизведения
NOLABELS	При значении TRUE вывод информации о проигрываемом файле RealAudio (автор, название и пр.) будет запрещен

Перечень возможных значений параметра CONTROLS (выбор элементов панели воспроизведения) приведен в табл. 7.3.

Таблица 7.3. *Значения параметра CONTROLS*

Значение	Описание
АИ	Значение по умолчанию — отображаются все элементы управления
`ControlPanel`	Кнопки воспроизведения и остановки, индикатор воспроизведения файла и регулятор громкости
`InfoPanel`	Вывод информации о файле
`InfoVolumePanel`	Вывод информации о файле и регулятор громкости
`StatusBar`	Окно состояния воспроизведения файла, указание времени и продолжительности файла
`PlayButton`	Только кнопка воспроизведения/паузы
`StopButton`	Только кнопка остановки
`VolumeSlider`	Только регулятор громкости
`PositionField`	Время воспроизведения и продолжительность файла в строке состояния
`PositionSlider`	Только индикатор текущего положения файла
`StatusField`	Текстовые сообщения в строке состояния

Фоновый звук

В завершение разговора о музыкальных форматах, используемых в современном Интернете, следует упомянуть о такой возможности языка разметки HTML, как указание фонового звука для электронного документа:

```
<BODY BGSOUND="audio/intro.wav" LOOP="infinite">
```

Данный пример задает для страницы фоновый звук `"intro.wav"` с помощью параметра `BGSOUND`. Конструкция `LOOP="infinite"` дает браузеру команду проигрывать указанный файл бесконечное количество раз. Для ограниченного количества воспроизведения фонового звука в качестве значения необходимо использовать целые числа (1, 2, 3 и т. д.).

Часть II

Создание интерактивных HTML-документов

6 Зак. 863

Глава 8

Работа с пользовательскими формами

Что такое пользовательские формы

Пользовательская форма — это совокупность стандартных HTML-конструкций ввода текстовой и прочей информации и программы-обработчика этой информации, работающей на Web-сервере. Иными словами, пользовательская форма (или HTML-форма) служит для передачи информационных данных серверу.

Результат конструкций языка разметки HTML интерпретируется браузером, с помощью которого пользователь электронного документа получает информацию. Таким образом, объединив все эти формулировки, можно сказать, что HTML-форма выступает в роли посредника между пользователем и сервером.

Посетитель Web-страницы вводит в HTML-форму определенные данные, которые обрабатываются программой и отсылаются на Web-сервер. Все эти действия укладываются в три стадии:

1. Ввод пользователем информации.
2. Обработка введенной информации программой, установленной на сервере.
3. Получение результата отправления введенной информации на Web-сервер (открытие нового HTML-документа, переадресация на предыдущую страницу и пр.).

В качестве программы-обработчика чаще всего выступает CGI-сценарий (скрипт, который обычно разрабатывается на языке Perl или C/C++ и который взаимодействует со специальным компонентом Web-сервера — Common Gateway Interface) или программы, написанные на основе таких серверных языков программирования, как PHP, ASP, JSP и др.

Значение пользовательских форм трудно переоценить — они являются особым средством HTML, дающим посетителю возможность не только пассивно просматривать информацию, но и быть задействованным в содержании

Web-сайта. Такое свойство принято называть интерактивностью, которая на сегодняшний день встречается практически во всех электронных документах.

Что же дает присутствие пользовательских форм на странице? Диапазон функций, выполняемых HTML-формами, чрезвычайно широк, поэтому остановимся лишь на некоторых, наиболее распространенных направлениях.

Организация обратной связи

Форма обратной связи — это неотъемлемый атрибут Web-сайта любой тематики и функциональной направленности. Размещая электронный документ во Всемирной сети, мы уже подразумеваем реакцию пользовательской аудитории — похвалу и одобрение или же несогласие и критику.

В качестве средства обратной связи между пользователем и автором электронного документа могут выступать:

- *почтовая форма* (рис. 8.1). Посетителя просят указать свои персональные данные и изложить суть обращения. Текст обращения обрабатывается программой и высылается автору сайта по электронной почте. При этом другие посетители сайта не могут получить доступ к содержанию отправленного сообщения;

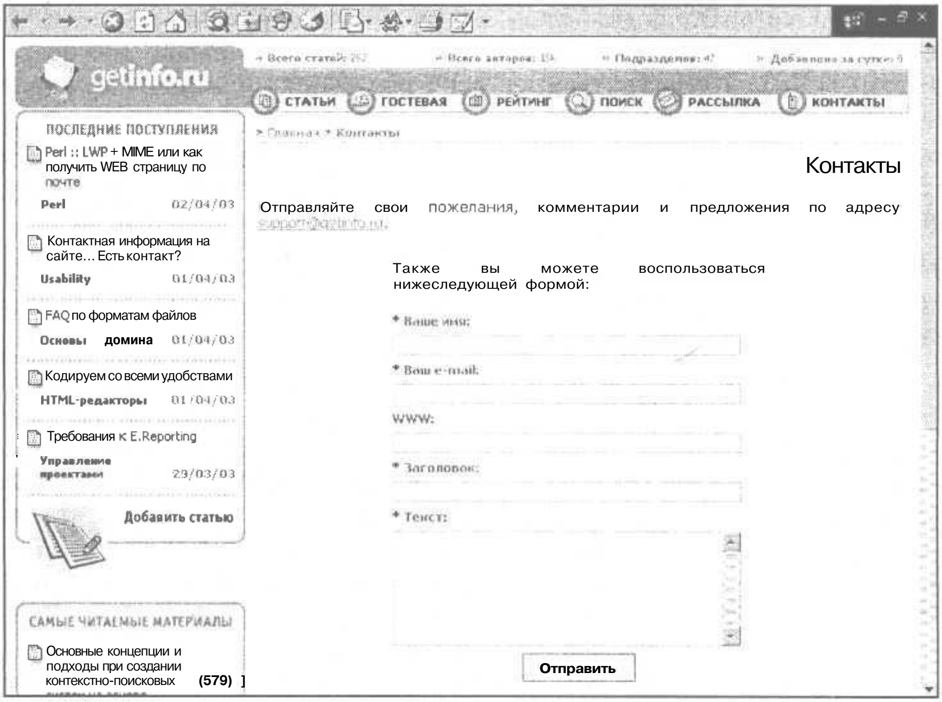


Рис. 8.1. Пример почтовой формы

- *гостевая книга* (рис. 8.2). Электронная книга жалоб и предложений, в которой любой желающий может оставить свой отзыв, поделиться впечатлениями о сайте, изложить критические замечания и указать на ошибки автора. В этом случае сообщения одного посетителя видимы для остальных пользователей гостевой книги;

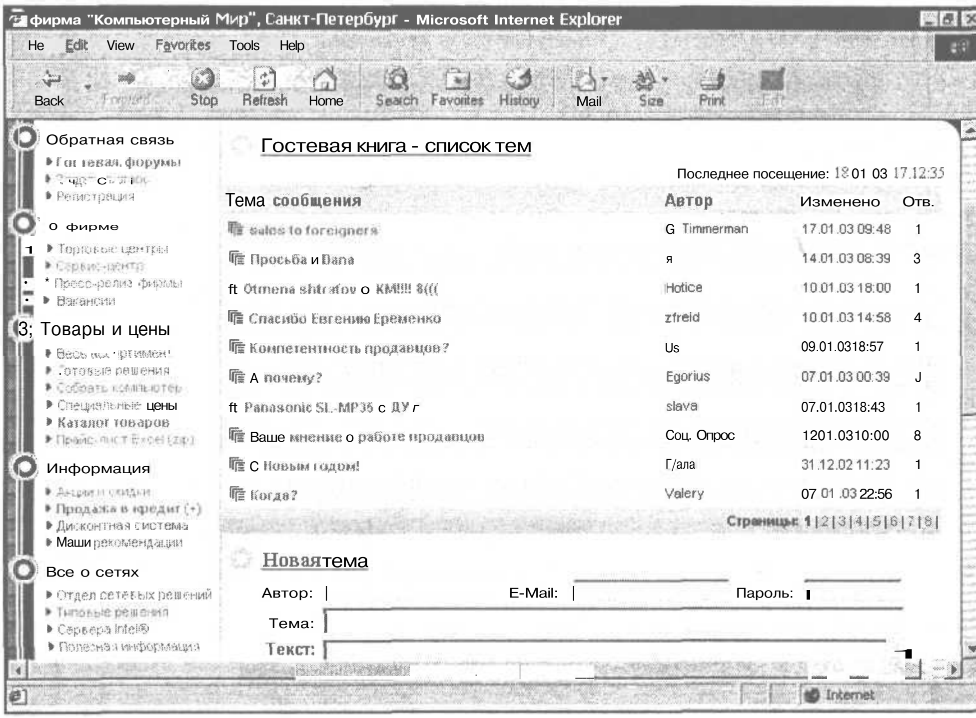


Рис. 8.2. Пример гостевой книги

- *интернет-форум* (рис. 8.3). Более мощный инструмент стимулирования оценки пользовательской аудитории, нежели почтовая форма или гостевая книга. В интернет-форумах происходит многосторонний диалог, причем не только между человеком, задающим вопрос, и автором Web-сайта, но и между всеми пользователями данного интернет-ресурса. На вопрос одного пользователя может ответить другой, третий подскажет то, чего не упомянул второй, наконец, автор сайта подытожит высказывания всех пользователей и даст свой развернутый ответ. Часто в рамках интернет-форума работают дополнительные функциональные средства общения: личные сообщения, скрытые разделы форума для привилегированных пользователей, опросы общественного мнения и пр.;

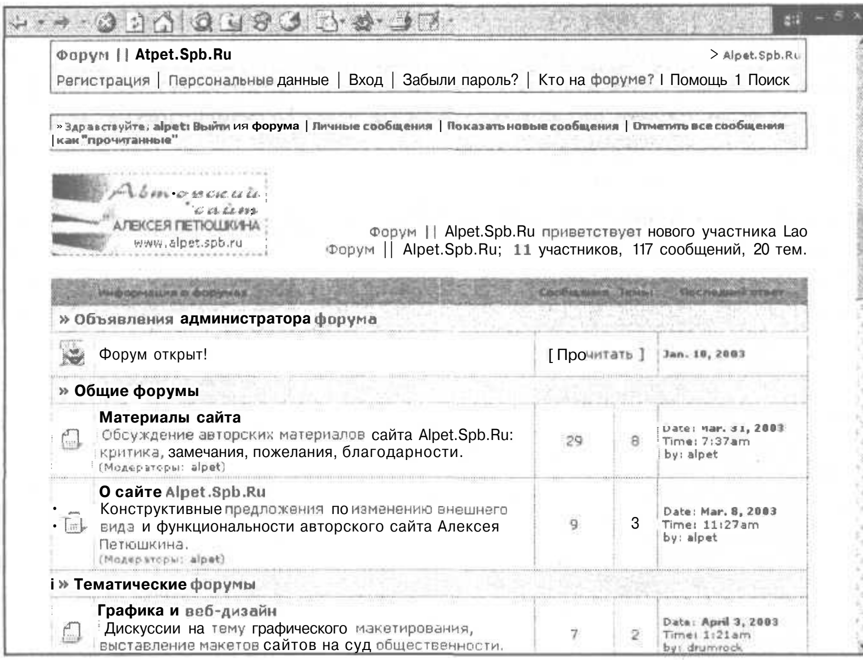


Рис. 8.3. Пример Интернет-форума

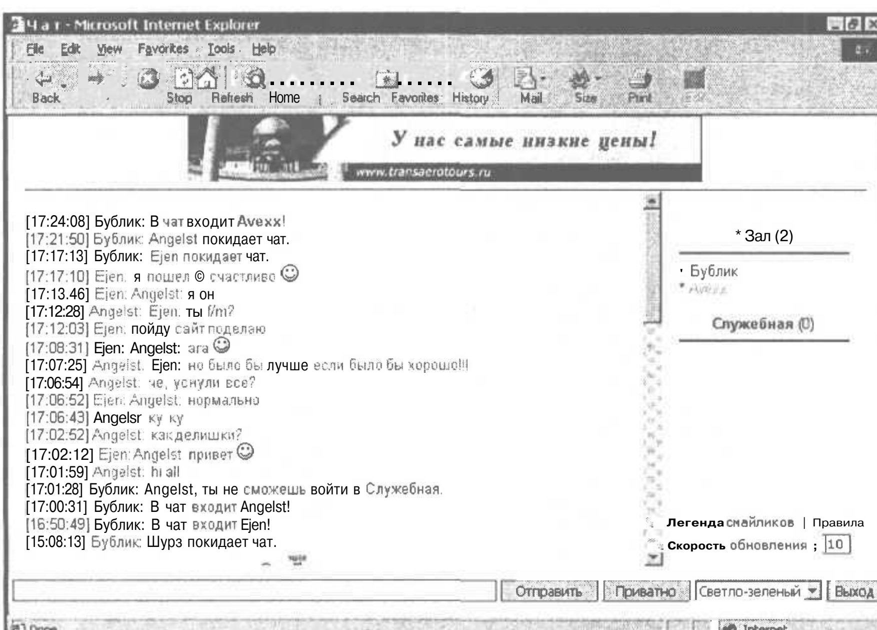


Рис. 8.4. Пример чата

- *чат* (рис. 8.4). Средство чата (от англ. "to chat" — разговаривать, болтать) подразумевает единовременное общение многих пользователей в режиме реального времени. В отличие от гостевой книги или форума, где ответ на поставленный вопрос может быть выслан и через час, и через день, общение в чате происходит между двумя (или более) собеседниками так, будто они сидят в одной комнате. Собеседники могут находиться в разных городах, государствах и временных зонах, однако присутствуя в чате, все пользователи способны общаться одновременно. В чатах также могут присутствовать некоторые дополнительные функции: приваты (отдельный разговор между двумя пользователями), система почтовых сообщений, голосования и т. д.

Авторизация

Для получения доступа к информации на многих интернет-ресурсах существует необходимость регистрации в сервисе. После регистрации, которая подразумевает указание уникального пользовательского имени и пароля, пользователь должен пройти процедуру авторизации, т. е. проверки на наличие в базе данных сайта заявленного имени и пароля (рис. 8.5).

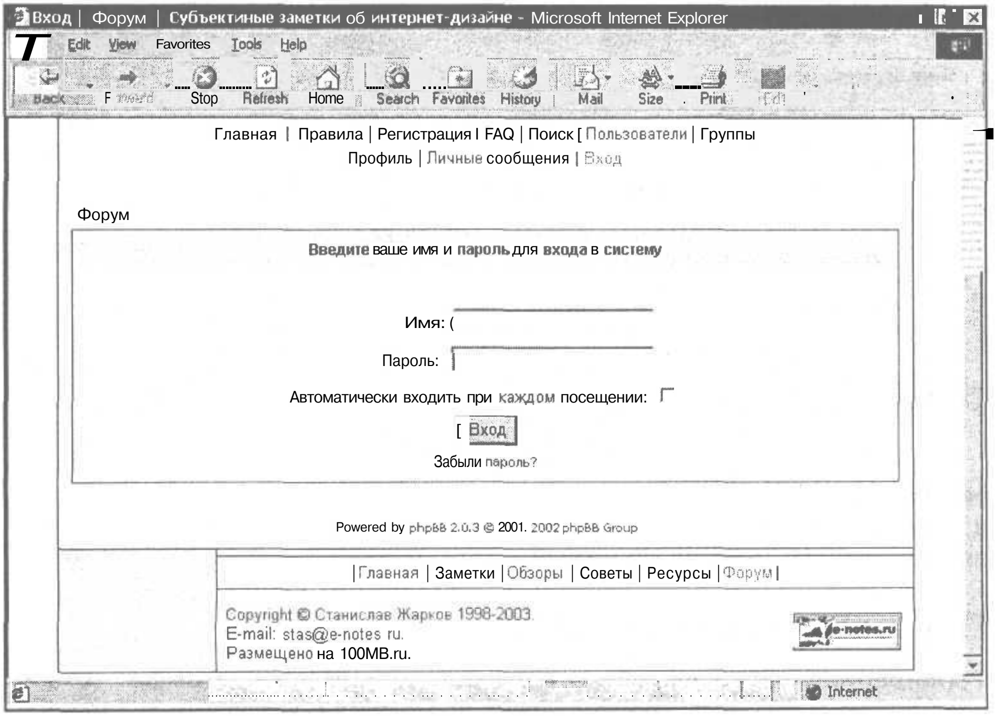


Рис. 8.5. Пример формы авторизации

Система авторизации распространена на информационных ресурсах (новостные ленты, афиши событий), корпоративных серверах (закрытые разделы для клиентов, партнеров, дистрибьюторов), сайтах электронной почты и т. д.

Проведение исследований

Для изучения пользовательского спроса на тот или иной вид услуг, выявления и последующего анализа реакции посетителей, на многих сайтах устраиваются всевозможные опросы, анкетирования, голосования, тесты и пр. (рис. 8.6).

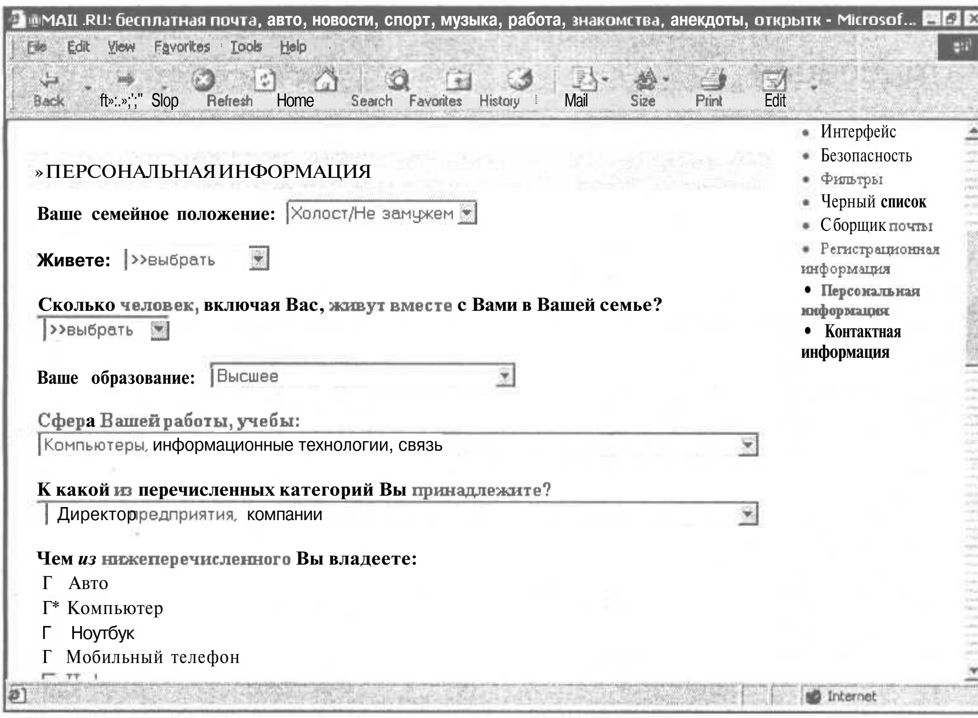


Рис. 8.6. Пример анкеты

Набор стандартных HTML-конструкций облегчает участие пользователя в подобных исследованиях. Все, что ему потребуется, это отметить подходящие варианты ответа и подтвердить участие в опросе.

Пользовательский профиль

Регистрируясь на каком-либо интернет-ресурсе, человек указывает всякого рода информацию: имя, город и страну проживания, увлечения, место работы, контактные данные и пр. Совокупность таких данных составляет поль-

зовательский профиль, который впоследствии возможно будет изменить путем редактирования соответствующих полей HTML-формы (рис. 8.7).

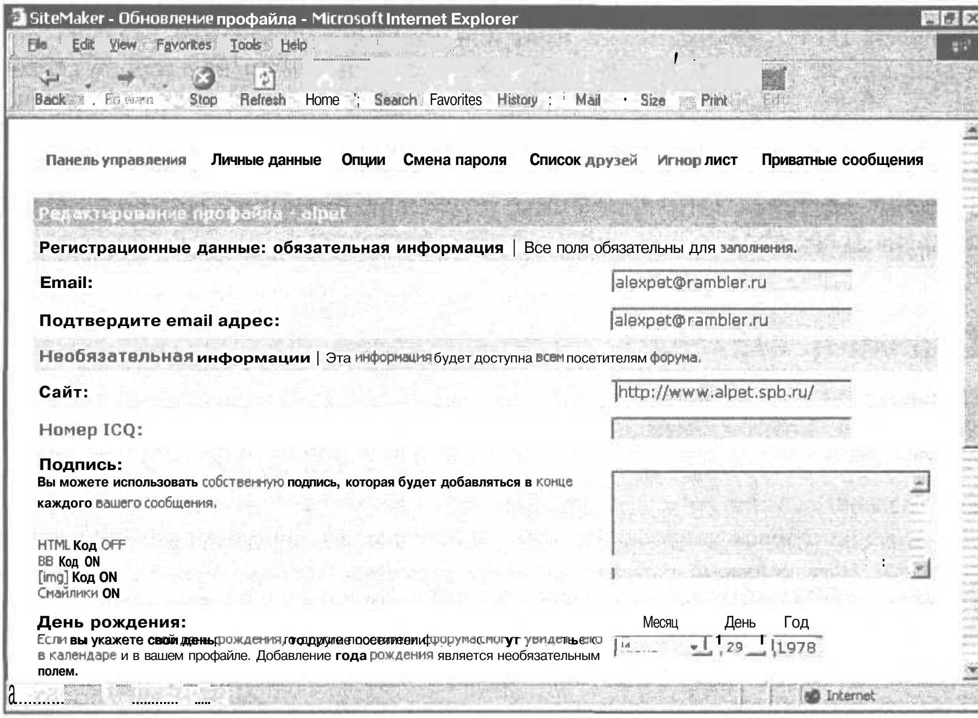


Рис. 8.7. Пример пользовательского профиля

Структура пользовательских форм

Структура любой пользовательской формы описывается тегом-контейнером <FORM>, внутри которого могут располагаться теги и параметры составляющих элементов HTML-формы. В пределах тега <FORM> не может содержаться другая пользовательская форма. Закрывающий тег обязателен для указания.

Тег <FORM> может содержать четыре основных параметра: ACTION, METHOD, NAME и ENCTYPE. Рассмотрим каждый из них.

Параметр *ACTION*

Параметр ACTION является единственным обязательным параметром тега <FORM> и предназначен для указания пути на Web-сервере к программе-обработчику данных пользовательской формы:

```
<FORM ACTION="/cgi-bin/form.cgi">
```

Путь к программе-обработчику может быть как относительный, так и абсолютный.

Значение этого параметра играет важную роль с точки зрения работоспособности HTML-формы. Если путь или название программы указаны неверно или же указанный файл не является исполняемым на стороне сервера, то обработка введенных пользователем данных может быть нарушена, произведена некорректно или вовсе не быть осуществленной.

Наиболее распространенными типами исполняемых на стороне сервера файлами являются CGI-программы (с расширениями pl, cgi, fcgi), PHP-скрипты (php, php3, phtm, phtml), ASP-приложения (asp), JSP-обработчики (jsp) и др.

Параметр *METHOD*

Параметр METHOD используется для определения типа передачи данных на Web-сервер. Возможными значениями параметра являются GET (по умолчанию) и POST.

При использовании типа GET данные пользовательской формы пересылаются в составе адреса запроса браузера: после имени программы-обработчика ставится знак вопроса (?), обозначающий вывод запроса браузера к переменным HTML-формы, а также последовательность переменных и их свойств из самой формы. Последовательность переменных формы разделяется символом амперсанда «&".

Пример адреса запроса браузера с использованием типа передачи данных GET:

```
http://www.site.ru/cgi-bin/form.cgi?Name=Vasya&Email=vasya@vasya.ru
```

Из структуры ссылки, образовавшейся в ходе обработки данных формы, понятно, что пользователь ввел свое имя («Vasya") и адрес электронной почты («vasya@vasya. ru").

При использовании типа POST данные формы не отображаются в адресной строке браузера, а передаются в составе самого запроса программы-обработчика. Таким образом, используя этот тип, мы получим следующую гиперссылку (учитывая те же данные формы, что и в случае с типом GET):

```
http://www.site.ru/cgi-bin/form.cgi
```

Следует отметить, что пользовательская форма может включать информацию о пользователе, носящую конфиденциальный характер (например, пароли доступа), которая будет отображена в запросе браузера при использовании типа GET. В этом случае вся секретная информация, введенная пользователем, будет доступна для просмотра любому пользователю данного компьютера (просмотр истории перехода по Web-сайтам в браузере легко обнаружитссылкусконфиденциальнымиданными).

Но в конечном итоге выбор типа передачи данных на Web-сервер определяется конкретной задачей, стоящей перед автором HTML-формы, особенностями сервера и самой программы-обработчика.

Параметр *NAME*

Параметр NAME присваивает HTML-форме уникальное имя, которое используется в программе-обработчике для идентификации пользовательских данных, например:

```
<FORM NAME="mail">
```

Параметр *ENCTYPE*

Параметр ENCTYPE предназначен для определения типа данных при кодировании информации, введенной пользователем, и последующей ее передаче на Web-сервер. Кодирование осуществляется браузером и используется для предотвращения разного рода искажений в процессе передачи на сервер.

Возможными значениями параметра могут быть: application/x-www-form-urlencoded (по умолчанию) и multipart/form-data.

Первое значение используется, если помимо текста необходимо передать на сервер данные иного типа (к примеру, графику или запакованные файлы). Формат записи состоит из указания типа и его подтипа. Тип данных — это определение общего типа данных (текст, графика, архив, программа и т. д.), например, text, image, application. Подтип — это вид данных внутри определенного общего типа (image/gif, text/html).

Значение multipart/form-data используется в редких специфических случаях, например, при необходимости предоставить пользователю возможность загрузки на сервер любого файла со своего локального компьютера.

Основные параметры тега <FORM> приведены в табл. 8.1.

Таблица 8.1. *Основные параметры тега <FORM>*

Параметр	Функция
ACTION	Указание пути на Web-сервере к программе-обработчику данных пользовательской формы
METHOD	Определение типа передачи пользовательских данных на Web-сервер
NAME	Указание уникального имени пользовательской формы для идентификации программой-обработчиком
ENCTYPE	Определение типа кодирования пользовательских данных при передаче от браузера к Web-серверу

Внутри тега <FORM> могут размешаться дополнительные теги и параметры, используемые для включения в состав пользовательской формы различных элементов управления. Они-то и определяют визуальное наполнение HTML-формы (тег <FORM> и его параметры не видимы пользователю и предназначены для указания необходимой информации браузеру и программе-обработчику на сервере).

Тег <*INPUT*>

Тег <INPUT> является наиболее распространенным и способен отобразить широкий набор элементов управления пользовательской формой:

- текстовую строку;
- поле ввода пароля;
- поле выбора локального файла для загрузки на Web-сервер;
- опцию выбора;
- опцию переключения;
- кнопку отправления пользовательских данных;
- графический вариант кнопки отправления пользовательских данных;
- кнопку сброса введенных пользовательских данных;
- скрытые поля.

Тег не является парным и может содержать параметры.

Параметр *TYPE*

Параметр TYPE позволяет указывать один из перечисленных элементов формы, каждому из которых соответствует отдельное значение.

`TYPE="TEXT"` — это элемент текстовой строки. Может содержать ряд дополнительных параметров, описанных в табл. 8.2.

Таблица 8.2. *Параметры текстовой строки* <INPUT TYPE="text">

Параметр	Описание
SIZE	Определение максимального количества символов, видимых в текстовой строке без перемещения курсора
MAXLENGTH	Определение максимального количества символов, допустимых для ввода в текстовой строке. По умолчанию количество вводимых символов не ограничено
NAME	Присвоение идентификационного имени для программы-обработчика
VALUE	Указание значения текстовой строки (при просмотре в браузере выводится в виде обычного текста в самой строке)

Рассмотрим несколько вариантов использования элемента текстовой строки в HTML-форме (листинг 8.1, рис. 8.8).

Листинг 8.1. Варианты использования элемента текстовой строки

```
<HTML>
  <HEAD>
    <TITLE>Варианты использования элемента текстовой строки</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H3>Пожалуйста, сообщите свои данные:</H3>
<FORM NAME="mail" ACTION="/cgi-bin/mail.cgi" METHOD="post">
Ваше имя:<BR>
<INPUT TYPE="text" SIZE="20" MAXLENGTH="50" NAME="name">
<BRXBR>
Ваш E-mail:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="email">
<BRXBR>
Ваш сайт:<BR>
<INPUT TYPE="text" SIZE="40" MAXLENGTH="60" NAME="www">
<BRXBR>
Комментарий:<BR>
<INPUT TYPE="text" SIZE="50" MAXLENGTH="100" NAME="comment"
VALUE="Краткий комментарий (не более 100 символов)">
</FORM>
</BODY>
</HTML>
```

TYPE="PASSWORD" — это элемент ввода пользовательского пароля. Ничем не отличается от обыкновенной текстовой строки, за исключением того, что набранный текст отображается в виде звездочек (рис. 8.9):

```
<INPUT TYPE="password" SIZE="30" NAME="password">
```

Такая мера связана с сохранением конфиденциальности пользовательских данных (однако не стоит забывать, что данные, вводимые в это поле, при использовании типа передачи GET будут отображаться в ссылке запроса браузера).

TYPE="FILE" -- это элемент выбора файла, расположенного на локальном компьютере пользователя, предназначенного для загрузки на Web-сервер (рис. 8.10). Из дополнительных параметров действуют только SIZE и MAXLENGTH.

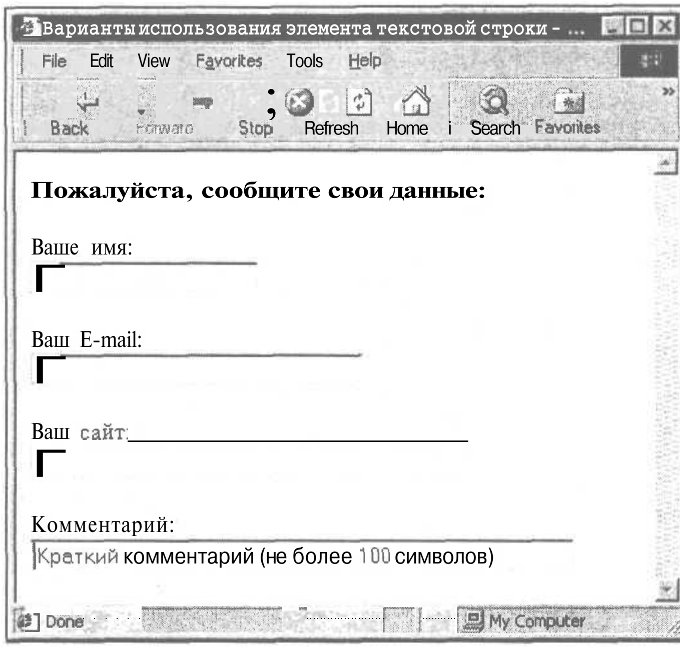


Рис. 8.8. Варианты использования элемента текстовой строки

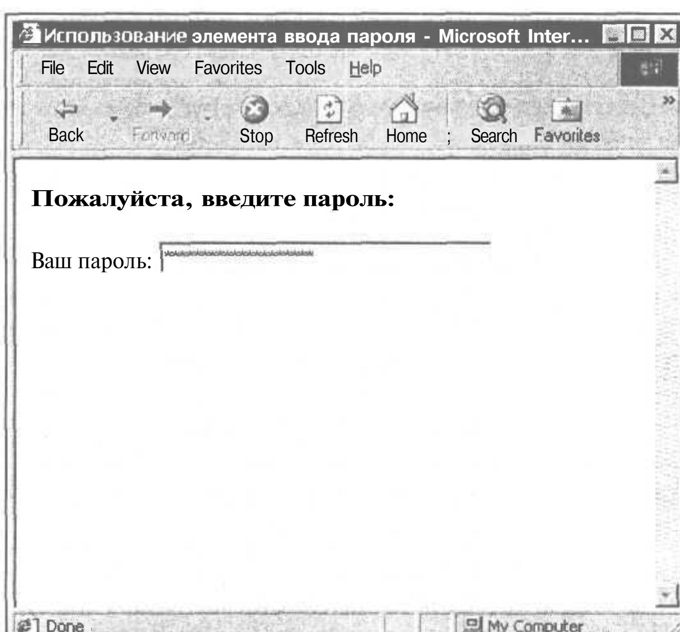


Рис. 8.9. Использование элемента ввода пароля

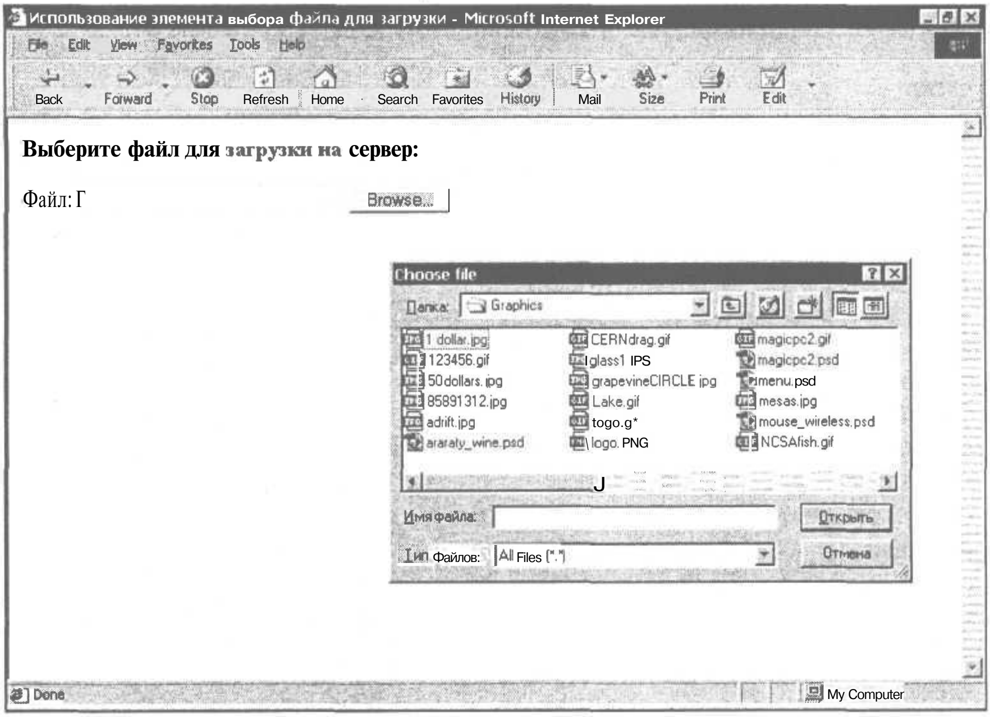


Рис. 8.10. Использование элемента выбора файла для загрузки

Для корректной передачи файла необходима конструкция `ENCTYPE="multipart/form-data"` и указание типа передачи данных POST в теге <FORM>. В противном случае браузер передаст не выбранный файл, а путь к нему на компьютере пользователя.

Следует добавить, что средств **HTML,** способных изменять название кнопки просмотра содержимого локального компьютера, не существует. Значение может быть только одно и зависит от модели браузера: "Browse" или "Обзор".

Элемент выбора локального файла ранними версиями популярных браузеров поддерживается не полностью. Например, Internet Explorer 3.*x* отображает только строку пути для файла, не давая возможности поиска файла на компьютере прямо из браузера (в этом случае на локальном компьютере пользователя приходится вручную вписывать полный путь к нужному файлу).

`TYPE="CHECKBOX"` — это элемент опции выбора, устанавливающий/снимающий флажок для конкретного поля пользовательской формы (листинг 8.2, рис. 8.11). Может содержать ряд дополнительных параметров, описанных в табл. 8.3.

Листинг 8.2. Использование элемента опции выбора

```
<HTML>
  <HEAD>
    <TITLE>Использование элемента опции выбора</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H3>Укажите свои увлечения:</H3>
<FORM NAME="select_hobby" ACTION="/cgi-bin/hobby.cgi" METHOD="get">
<INPUT TYPE="checkbox" NAME="hobby" VALUE="computers" CHECKED>Компьютеры
и Интернет
<BRXBR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="art">Литература и искусство
<BRXBR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="music">Музыка и танцы
<BRXBR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="auto">Автомобили
<BRXBR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="sport">Спорт и активный отдых
</FORM>
</BODY>
</HTML>
```

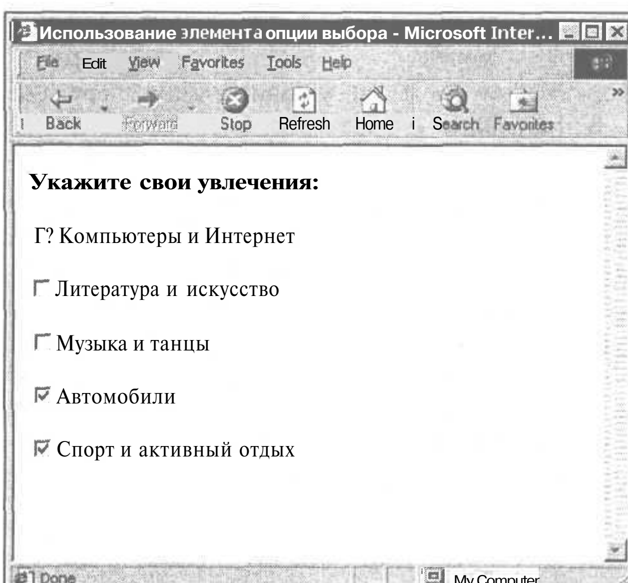


Рис. 8.11. Использование элемента опции выбора

Таблица 8.3. Параметры опции выбора *<INPUT TYPE="checkbox">*

Параметр	Описание
NAME	Указание общего для всех вариантов выбора идентификационного имени
VALUE	Определение значения для конкретного варианта выбора (обязательный параметр). Не должен повторяться, т. к. при установке флажка передается на Web-сервер
CHECKED	Данный вариант является выбранным по умолчанию

Как видно из рисунка, пользователь может выбирать несколько вариантов поля CHECKBOX, значение каждого из которых будет передано программой-обработчиком на Web-сервер.

TYPE="RADIO" — это элемент опции переключения между различными вариантами выбора (листинг 8.3, рис. 8.12). В отличие от CHECKBOX, вариант выбора может быть только один. Также имеет дополнительные параметры NAME, VALUE и CHECKED, действие которых аналогично опции выбора CHECKBOX.

Листинг 8.3. Использование элемента опции переключения

```
<HTML>
  <HEAD>
    <TITLE>Использование элемента опции переключения</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H3>Укажите свою возрастную группу:</H3>
<FORM NAME="select_age" ACTION="/cgi-bin/age.cgi" METHOD="get">
<INPUT TYPE="radio" NAME="age" VALUE="baby" CHECKED>0-5 лет
<BRXBR>
<INPUT TYPE="radio" NAME="age" VALUE="child">5-12 лет
<BRXBR>
<INPUT TYPE="radio" NAME="age" VALUE="junior">12-20 лет
<BRXBR>
<INPUT TYPE="radio" NAME="age" VALUE="adult">от 20 лет
</FORM>
</BODY>
</HTML>
```

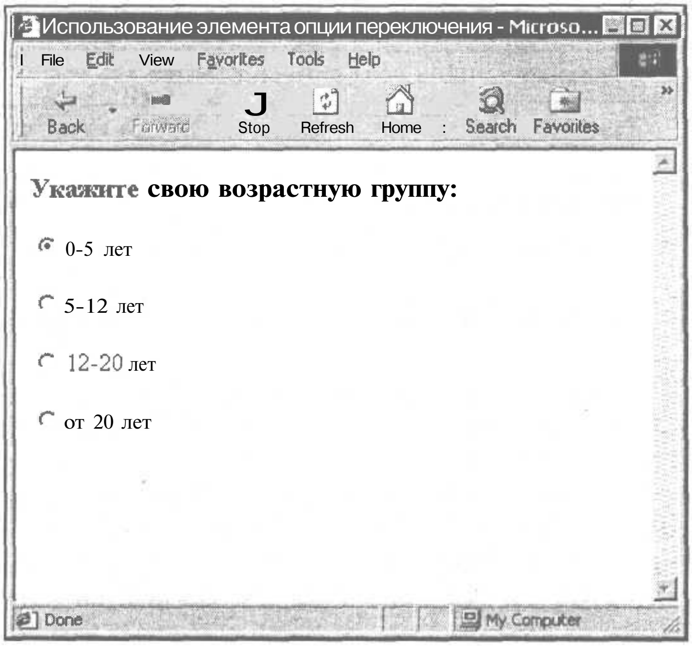


Рис. 8.12. Использование элемента опции переключения

`TYPE="SUBMIT"` — это кнопка отправления пользовательских данных на Web-сервер. При нажатии на нее запускается программа-обработчик, которая анализирует введенные пользователем данные и отправляет результат на сервер.

В принципе, необязательный элемент. Отправление данных может быть выполнено нажатием клавиши <Enter> на клавиатуре пользователя при размещении курсора мыши на любом поле формы. Однако подобный подход не приветствуется, т. к. отсутствие кнопки подтверждения может запутать неопытного пользователя.

По умолчанию отображается в виде прямоугольной кнопки с надписью "Submit" ("Submit Query" и пр., в зависимости от браузера).

Для элемента отправления данных на сервер также могут использоваться дополнительные (необязательные) параметры — `NAME` (имя кнопки отправления, значение которого также будет передано на сервер) и `VALUE` (присвоение собственного имени кнопке).

`TYPE="IMAGE"` — это графический аналог стандартной кнопки отправления данных формы на Web-сервер. В этом случае автор электронного документа не ограничен средствами HTML и может применить всю свою фантазию на создание оригинальной, привлекательной кнопки передачи данных.

Для данного параметра используются некоторые дополнительные параметры:

- SRC — указание относительного или абсолютного пути на сервере к файлу графического изображения, служащего кнопкой передачи данных формы;
- ALIGN — указание типа выравнивания текста относительно графической кнопки отправления данных формы;
- BORDER — определение толщины рамки кнопки (как правило, значение равно нулю);
- ALT — указание альтернативного текста/подсказки для кнопки отправления данных.

TYPE="RESET" — это кнопка сброса введенных пользователем данных HTML-формы. При нажатии на нее восстанавливаются все установленные по умолчанию значения полей формы. Элемент не обязателен, однако в ряде случаев весьма полезен. При работе с многочисленными текстовыми строками и опциями выбора пользователь может, допустив ошибку, пожелать заново заполнить форму, тогда ему придется либо перезагружать страницу, либо вручную удалять текст из каждого поля формы. Кнопка сброса в этом случае позволит добиться желаемого при одном нажатии на нее. Пример использования кнопок отправления и сброса данных приведен соответственно в листинге 8.4 и на рис. 8.13.

По умолчанию названием кнопки сброса является "Reset". Изменить наименование элемента можно с помощью дополнительного параметра VALUE.

Листинг 8.4. Использование кнопок отправления и сброса данных

```
<HTML>
  <HEAD>
    <TITLE>Использование кнопок отправления и сброса данных</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<FORM NAME="user_data" ACTION="/cgi-bin/anketa.cgi" METHOD="post">
<H2>Анкета пользователя:</H2>
<P><B>Контактные данные:</B></P>
Ваше имя:<BR>
<INPUT TYPE="text" SIZE="20" MAXLENGTH="50" NAME="name">
<BR>
Ваш E-mail:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="email">
<BR>
```

```
Ваш сайт:<BR>
<INPUT TYPE="text" SIZE="40" MAXLENGTH="60" NAME="www">
<P><B>Возраст:</BX/P>
<INPUT TYPE="radio" NAME="age" VALUE="baby" CHECKED>0-5
лет  <INPUT TYPE="radio" NAME="age" VALUE="child">5-12
лет    <INPUT TYPE="radio" NAME="age"
VALUE="junior">12-20 лет  <INPUT TYPE="radio" NAME="age"
VALUE="adult">от 20 лет
<P><B>Увлечения:</BX/P>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="computers" CHECKED>Компьютеры
и Интернет  <INPUT TYPE="checkbox" NAME="hobby"
VALUE="art">Литература и искусство  <INPUT TYPE="checkbox"
NAME="hobby" VALUE="music">Музыка и танцы  <INPUT
TYPE="checkbox" NAME="hobby" VALUE="auto">Автомобили  <INPUT
TYPE="checkbox" NAME="hobby" VALUE="sport">Спорт и активный отдых
<P ALIGN ="center">
<INPUT TYPE="submit" VALUE="Отправить данные"> <INPUT TYPE="reset"
VALUE="Очистить поля формы">
</P>
</FORM>
</BODY>
</HTML>
```

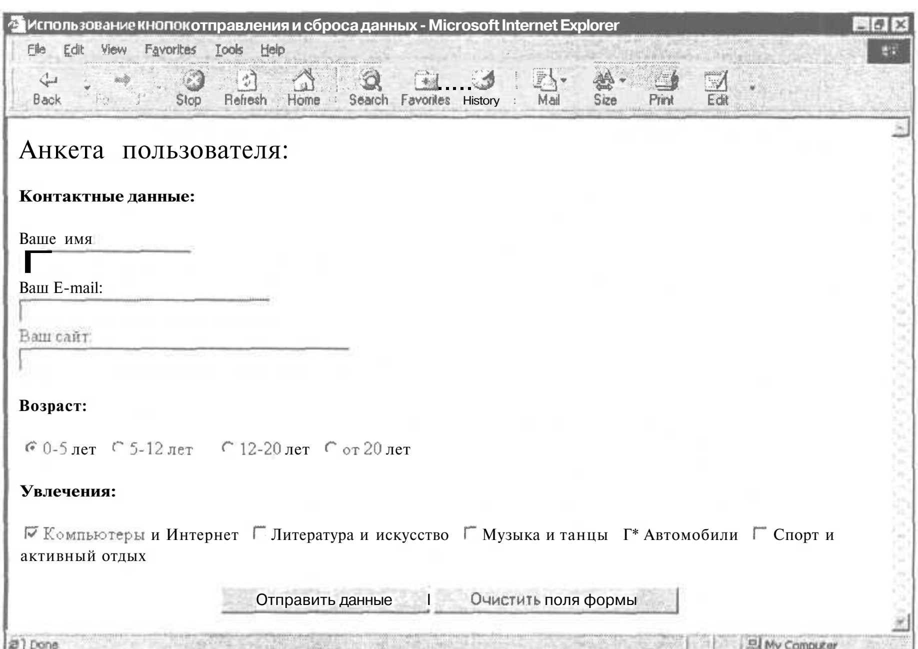


Рис. 8.13. Использование кнопок отправления и сброса данных

TYPE="HIDDEN" — это служебное поле, не видимое пользователю, не может быть изменено ни браузером, ни самим пользователем. Анализируется программой-обработчиком и отсылается на Web-сервер (отсылаемое значение варьируется специальным параметром VALUE).

Дополнительные параметры тега <*INPUT*>

Кроме основных параметров, в теге <INPUT> могут присутствовать некоторые дополнительные параметры, поддерживаемые не всеми браузерами и используемые с помощью вспомогательных средств по отношению к HTML. К ним можно отнести следующие параметры:

- DISABLED - параметр, отменяющий активность выбранного поля формы. При указании параметра DISABLED редактирование элемента формы будет недоступно. Параметр поддерживается только браузером Internet Explorer. Используется в сложных, динамически изменяющихся HTML-формах в совокупности с такими технологическими средствами, как DHTML, CSS, JavaScript и др.;
- TITLE — параметр отображения всплывающей подсказки для элемента формы. Поддерживается только браузерами: Internet Explorer *(4.x* и выше), Netscape (6.x и выше), Opera (6.0 и выше);
- STYLE — параметр определения стиля CSS для выбранного элемента формы;
- CLASS - параметр, устанавливающий класс стилевого шаблона CSS (ссылающийся на внешний или внутренний набор стилей).

Тег <*TEXTAREA*>

Тег-контейнер <TEXTAREA> создает элемент текстового поля заданной ширины и высоты (листинг 8.5, рис. 8.14). Указание закрывающего тега обязательно. Внутри тега <TEXTAREA> не может быть никаких других тегов или параметров.

Главное отличие элемента текстового поля от элемента текстовой строки заключается в визуальном преимуществе объема вводимой информации. Основные параметры тега <TEXTAREA> приведены в табл. 8.4.

Таблица 8.4. *Параметры текстового поля <TEXTAREA>*

Параметр	Описание
COLS	Определение количества столбцов текстового поля
ROWS	Определение количества рядов текстового поля
NAME	Присвоение уникального имени, необходимого для идентификации программой-обработчиком

Листинг 8.5. Использование текстового поля

```
<HTML>
  <HEAD>
    <TITLE>Использование текстового поля</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H3>Пожалуйста, сообщите свои данные:</H3>
<FORM NAME="mail" ACTION="/cgi-bin/mail.cgi" METHOD="get">
Ваше имя:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="name">
<BR>
Ваш E-mail:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="email">
<BR>
Комментарий:<BR>
<TEXTAREA COLS="25" ROWS="5" NAME="comment">Ваш комментарий...</TEXTAREA>
<BRXBR>
<INPUT TYPE="submit" VALUE="Отправить"> <INPUT TYPE="reset"
VALUE="Очистить">
</FORM>
</BODY>
</HTML>
```

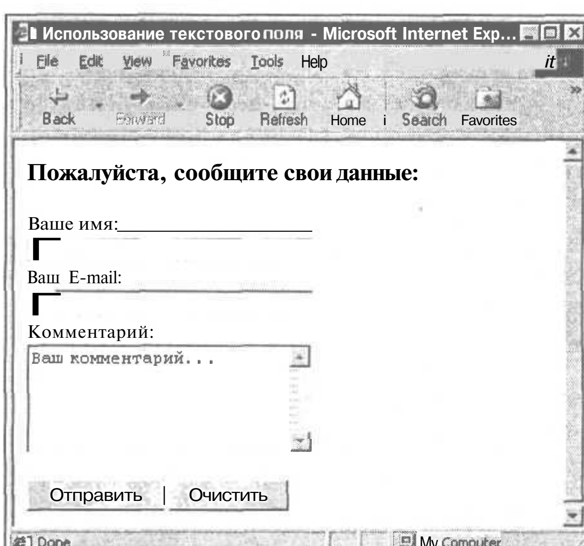


Рис. 8.14. Использование текстового поля

Тег <SELECT>

Тег <SELECT> предназначен для компактной группировки большого количества элементов пользовательской формы. К примеру, размещение нескольких десятков элементов CHECKBOX или RADIO займет слишком много места на странице, тогда как группировка данных с помощью тега <SELECT> позволяет заметно сократить размеры, занятые под отображение HTML-формы.

Такой вид формы может быть представлен ниспадающим меню или списком наименований (листинг 8.6, рис. 8.15).

Листинг 8.6. Использование ниспадающего меню и списка наименований

```
<HTML>
  <HEAD>
    <TITLE>Использование ниспадающего меню и списка наименований</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<FORM NAME="anketa" ACTION="/cgi-bin/anketa.cgi" METHOD="get">
<H2>В каком городе вы живете?</H2>
<SELECT NAME="city" SIZE="5">
 <OPTION VALUE="Moscow">Москва
 <OPTION VALUE="Spb" SELECTED>Санкт-Петербург
 <OPTION VALUE="Samara">Самара
 <OPTION VALUE="Ryazan">Рязань
 <OPTION VALUE="Krasnoyarsk">Красноярск
 <OPTION VALUE="Komsomolsk">Комсомольск-на-Амуре
 <OPTION VALUE="Vladivostok">Владивосток
</SELECT>
<H2>Ваша профессия?</H2>
<SELECT NAME="profession">
 <OPTION VALUE="Teacher">Учитель
 <OPTION VALUE="Technic">Слесарь
 <OPTION VALUE="Designer">Дизайнер
 <OPTION VALUE="Driver" SELECTED>Водитель
 <OPTION VALUE="Fireman">Пожарный
 <OPTION VALUE="Actor">Актер
</SELECT>
<BR><BR>
```

```
<INPUT TYPE="submit" VALUE="Отправить"> <INPUT TYPE="reset"
VALUE="Очистить">
</FORM>
</BODY>
</HTML>
```

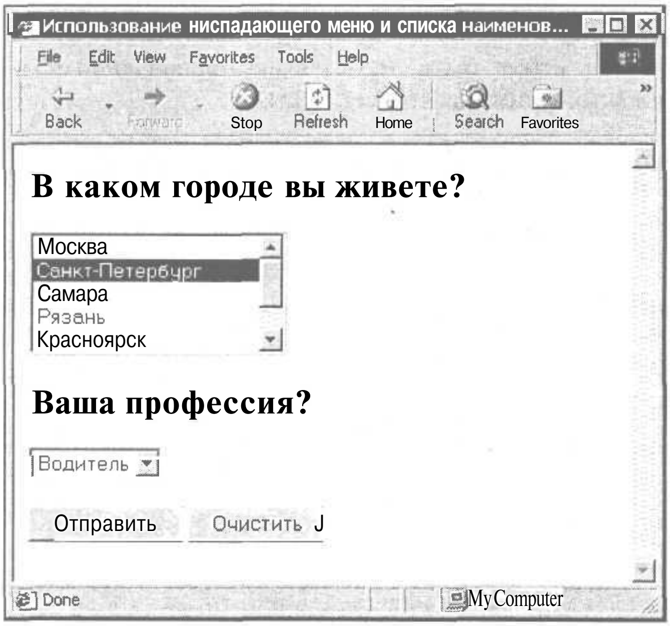


Рис. 8.15. Использование ниспадающего меню и списка наименований

Тег-контейнер <SELECT> определяет структуру и вид группировки элементов формы, требует закрывающего тега и может включать в себя следующие параметры:

- NAME -- уникальное имя, предназначенное для идентификации программой-обработчиком. Является обязательным параметром, значение которого передается на Web-сервер;
- SIZE — параметр, значение которого определяет число позиций ниспадающего меню, состоящего из списка наименований;
- MULTIPLE — параметр, разрешающий выбор нескольких позиций из списка наименований (выбор осуществляется с помощью курсора мыши при нажатой клавише <Ctrl>).

Внутри тега-контейнера <SELECT> не может находиться никакой информации, а также прочих тегов и их параметров, за исключением тега <OPTION>, который задает свойства для каждой из позиций ниспадающего меню или целого списка наименований.

В теге-контейнере <OPTION> могут быть размещены два основных параметра:

- VALUE — параметр, значение которого передается программой-обработчиком на Web-сервер. В данном случае будет отправлено значение конкретной выбранной позиции, а не общего значения меню или списка. Следует отметить, что при отсутствии значения параметра VALUE на сервер будет передано содержимое выбранной позиции (т. е. тега-контейнера <OPTION>);
- SELECTED - - параметр отмечает текущую позицию ниспадающего меню или списка наименований как выбранную. Таким образом возможно сделать визуальный акцент на любой по счету позиции меню или списка.

Следует отметить, что тег <OPTION> не требует обязательного присутствия закрывающего тега.

В табл. 8.5 приведены описанные теги и параметры, используемые для организации пользовательских форм.

Таблица 8.5. *Теги и параметры пользовательских форм*

Тег	Параметры	Описание
FORM	ACTION METHOD NAME ENCTYPE	Основной тег структуры пользовательской формы. Определяет путь на Web-сервере к программе-обработчику, тип передачи данных и кодирования пересылаемой информации, а также присваивает уникальное имя форме для программы-обработчика
INPUT	TYPE NAME VALUE SIZE CHECKED MAXLENGTH	Элемент текстовой строки, поле ввода пароля, опции выбора/переключения, подтверждения/отмены отправления данных формы, загрузки локального файла на Web-сервер, определение скрытых полей формы
TEXTAREA	COLS ROWS NAME	Элемент текстового поля
SELECT	NAME SIZE MULTIPLE	Группировка данных пользовательской формы путем составления ниспадающих меню и списков наименований
OPTION	VALUE SELECTED	Элемент (позиция) ниспадающего меню (списка наименований)

В завершение этой главы рассмотрим итоговый пример организации пользовательских форм с применением таблиц, текстового и структурного форматирования (листинг 8.7, рис. 8.16).

Листинг 8.7. Итоговый пример по организации пользовательских форм

```
<HTML>
  <HEAD>
    <TITLE>Итоговый пример по организации пользовательских форм</TITLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H2 ALIGN="center">ПОЛЬЗОВАТЕЛЬСКАЯ АНКЕТА</H2>
<FORM NAME="user_anketa" ACTION="/cgi-bin/anketa.cgi" METHOD="post"
ENCTYPE="multipart/form-data">
<TABLE BORDER="0" CELLPADDING="5" CELLSPACING="3" ALIGN="center"
WIDTH="98%">
<TR>
<TH BGCOLOR="#999999" WIDTH="50%">Личные данные</TH>
<TH BGCOLOR="#999999" WIDTH="50%">Увлечения</TH>
</TR>
<TR>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
Ваше имя:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="name">
<BR>
Ваш пароль:<BR>
<INPUT TYPE="password" SIZE="30" MAXLENGTH="35" NAME="password">
<BR>
Ваша фотография:<BR>
<INPUT TYPE="file" SIZE="18" NAME="photo">
</TD>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
<INPUT TYPE="checkbox" NAME="hobby" VALUE="computers" CHECKED>Компьютеры
и Интернет
<BR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="art">Литература и искусство
<BR>
```

```
<INPUT TYPE="checkbox" NAME="hobby" VALUE="music">Музыка и танцы
<BR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="auto">Автомобили
<BR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="sport">Спорт и активный отдых
</TD>
</TR>
<TR>
<TH BGCOLOR="#999999" WIDTH="50%">Место жительства</TH>
<TH BGCOLOR="#999999" WIDTH="50%">Комментарии</TH>
</TR>
<TR>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
<SELECT NAME="city" SIZE="5">
 <OPTION VALUE="">---Выберите город ---
 <OPTION VALUE="Moscow">Москва
 <OPTION VALUE="Spb">Санкт-Петербург
 <OPTION VALUE="Samara">Самара
 <OPTION VALUE="Ryazan">Рязань
 <OPTION VALUE="Krasnoyarsk">Красноярск
 <OPTION VALUE="Komsomolsk">Комсомольск-на-Амуре
 <OPTION VALUE="Vladivostok">Владивосток
</SELECT>
<BRXBR>
<SELECT NAME="country">
 <OPTION VALUE="">---Выберите страну ---
 <OPTION VALUE="Russia">Россия
 <OPTION VALUE="England">Англия
 <OPTION VALUE="USA">США
 <OPTION VALUE="Germany">Германия
 <OPTION VALUE="France">Франция
</SELECT>
</TD>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
<TEXTAREA COLS="25" ROWS="5" NAME="comments"></TEXTAREA>
<BR>
```

```
<INPUT TYPE="radio" NAME="site" VALUE="first" CHECKED>Я впервые на этом
сайте
<BR>
<INPUT TYPE="radio" NAME="site" VALUE="not_first">Я тут не в первый раз
</TD>
</TR>
<TR>
<TD COLSPAN="2" ALIGN="center" BGCOLOR="#999900">
<INPUT TYPE="submit" VALUE="Отправить"> <INPUT TYPE="reset"
VALUE="Очистить">
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

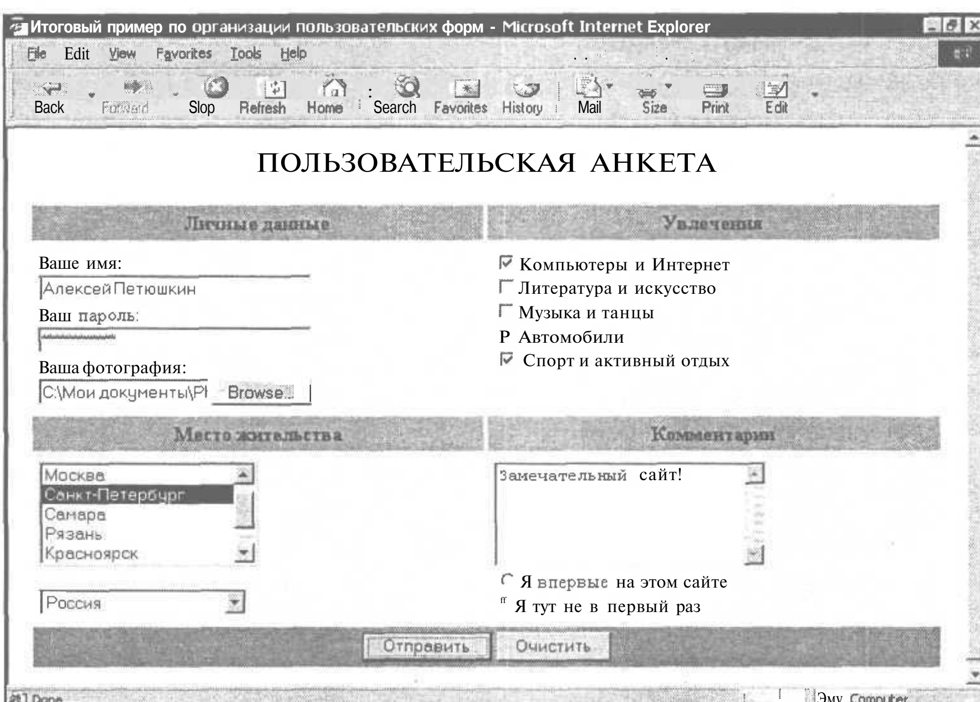


Рис. 8.16. Итоговый пример по организации пользовательских форм

В данном примере были использованы следующие элементы пользовательских форм:

- текстовая строка;
- поле для ввода пароля;
- поле выбора локального файла для загрузки на Web-сервер;
- опция выбора нескольких вариантов данных;
- опция переключения между несколькими вариантами данных;
- текстовое поле;
- ниспадающее меню;
- список наименований;
- кнопка отправления данных;
- кнопка сброса данных.

Глава 9

CSS и Dynamic HTML — просто и привлекательно!

Каскадные таблицы стилей

Спецификация языка разметки HTML позволяет разработчику электронных документов изменять внешний вид некоторых элементов страниц. Для этого составляются специальные правила отображения конкретного элемента в HTML-документе, называемые каскадными таблицами стилей (Cascading Style Sheets, CSS) или стилевыми шаблонами.

Разберем по составу это понятие — каскадная таблица стилей:

- каскадная. Спецификация HTML разрешает использовать для одного и того же элемента несколько стилевых правил, интерпретируемых браузером последовательно, другими словами — каскадом;
- таблица. Формат записи стилевых правил CSS напоминает табличное представление данных. Заголовок таблицы соответствует наименованию элемента, класса или идентификатора стиля. В качестве ячеек и рядов таблицы выступают стилевые свойства и их значения. Судите сами:

  ```
  DIV
  {
  font-family: Tahoma;
  color: black;
  font-size: 12px;
  }
  ```
- стиль. Под стилем принято понимать приведение какого-то явления к общему набору правил и определений. CSS — это способ дополнительного форматирования стандартных тегов HTML.

Версия HTML 4.01 включает следующие характеристики CSS:

- гибкое использование стилевых шаблонов (Flexible placement of style information). Правила задания стилевых таблиц позволяют использовать такие способы определения стиля, как создание внешнего CSS-файла,

использование внутреннего шаблона и импортирования таблицы стилей и пр. Более подробно о способах определения таблиц стилей будет рассказано *в разделе "Способы определения таблиц стилей" этой главы;*

- независимость от других языков определения стиля (Independence from specific style sheet languages). Спецификация HTML позволяет, помимо CSS, использовать другие языки определения стилевых шаблонов, что делает создаваемый электронный документ стиленезависимым, а значит и более адаптируемым к изменению внешнего вида определенных HTML-элементов;
- каскадность (Cascading). Возможность определения нескольких стилевых правил (указания нескольких таблиц стилей) для одного элемента HTML. Не все языки таблиц стилей поддерживают каскадность. Свойство каскадное™ реализуется за счет последовательного чтения стилевых данных браузером;
- привязанность к носителю информации (Media dependencies). С помощью стилевых шаблонов CSS возможна ориентация создаваемого электронного документа на различные носители информационных данных: операционные системы Windows и Macintosh, телевизионные и игровые приставки, мобильные телефоны и карманные персональные компьютеры (КПК), устройства распознавания человеческой речи и пр;
- альтернативные стили (Alternate styles). Разработчики электронных документов могут создавать несколько вариантов отображения элементов HTML с помощью различных таблиц стилей CSS.

Уровни CSS

Впервые каскадные таблицы стилей CSS были реализованы в браузере Internet Explorer 3.0. Однако в то время развитие CSS находилось в зачаточном состоянии, поэтому правила составления стилевых шаблонов были весьма разрозненными.

С момента своего возникновения структура CSS была несколько раз пересмотрена, в нее были добавлены новые элементы и убраны (видоизменены) старые. Существуют три уровня CSS, определяемых наличием завершенной редакции структуры. Это: CSS 1 (первый уровень структуры стилевых шаблонов, окончательно утвержденный 11 января 1999 года), CSS 2 (второй уровень стилевых конструкций, начало обсуждения которого датируется маем 1998 года) и CSS 3 (третий уровень стилевого оформления электронных документов, принятый к обсуждению 23 мая 2001 года, на момент написания книги находился в стадии проработки).

В завершение разговора об уровнях CSS следует добавить, что переход от одного уровня к другому, в основном, сопровождался некоторыми видоизменениями в структуре и в правилах стилевого оформления, технологическими дополнениями, а также попытками систематизировать применение CSS.

Именно третий уровень (CSS 3) позиционируется разработчиками в качестве некой единой системы представления стилей в электронном документе, основанной на использовании специальных модулей.

Способы определения таблиц стилей

Как уже было сказано, любая таблица стилей CSS должна быть интерпретирована браузером для того, чтобы правила CSS, обозначенные для конкретных элементов электронного документа, вступили в силу.

Определение таблицы стилей (стилевого шаблона) возможно четырьмя способами:

- *ссылка на внешний файл.* Если все стилевые шаблоны для конкретного HTML-документа разместить в одном текстовом файле (с расширением css), то с помощью специального тега <LINK> из текущего документа можно сделать ссылку на внешний CSS-файл стилевых шаблонов, например:

  ```
  <LINK REL="stylesheet" TYPE="text/css" HREF="style.css">
  ```

  Браузер, анализируя HTML-код, обратится по указанному пути и, обнаружив указанный файл стилевого оформления, отобразит элементы страницы в соответствии с определенными правилами CSS.

  Следует помнить, что конструкция указания пути к внешнему CSS-файлу должна находиться в пределах раздела HEAD HTML-документа;
- *внедрение в документ.* Под внедрением в документ подразумевается задание стилевой конструкции внутри самой HTML-страницы, например:

  ```
  <STYLE TYPE="text/css">
  <!—
  BODY { font-family: Arial, Helvetica; }
  INPUT { background-color: #CECECE; }
  -->
  </STYLE>
  ```

  Данная конструкция также должна присутствовать в разделе HEAD. Для браузеров, не поддерживающих CSS вообще или поддерживающих лишь отдельные правила стилевого оформления, описание шаблонов заключается между символами комментариев (при отсутствии поддержки CSS браузер пропустит содержание стилевых шаблонов, если же поддержка есть, то браузер интерпретирует правила CSS);
- *включение в теговые конструкции.* Любой отдельный HTML-элемент может быть подвергнут форматированию средствами CSS. Для этого необходимо задать определенное правило реализации того или иного тега, например:

  ```
  <P ALIGN="justify" STYLE="color: #000000; font-family: Verdana;">
  Текст параграфа...
  </P>
  ```

В этом случае задано отдельное правило для конкретного параграфа. Также можно присваивать отдельному HTML-элементу определенный класс стилевого шаблона:

```
<TABLE>
<TR>
<TD CLASS="header"></TD>
<TD CLASS="text"></TD>
</TR>
</TABLE>
```

Описание классов должно строиться следующим способом (на примере внедрения стилевого шаблона в документ):

```
<STYLE TYPE="text/css">
<!—
.header { font-weight: bold; color: gray; }
.text { color: black; font-size: 11px; }
-->
</STYLE>
```

В данном случае текст табличной ячейки класса .header будет отображаться жирным начертанием и серым цветом, а ячейки класса .text — обычным начертанием, черным цветом и размером шрифта 11 пикселов;

- *импортирование*. Импортирование стилевого шаблона CSS, по сути, аналогично указанию ссылки на внешний файл:

```
<STYLE TYPE="text/css">
<!—
@import: url(style.css);
-->
</STYLE>
```

Все четыре способа определения стилевого шаблона CSS можно использовать одновременно в пределах одного HTML-документа (листинг 9.1). Такая возможность позволяет задавать основное правило CSS, к примеру, в виде внешнего файла шаблонов, а для исключительных или редких HTML-элементов — отдельные конструкции либо в теге <STYLE>, либо в кодовых конструкциях самих тегов.

Листинг 9.1. Совмещение различных способов определения стилевого шаблона CSS

```
<HTML>
  <HEAD>
    <TITLE>Совмещение различных способов определения CSS</TITLE>
```

```
    <LINK REL="stylesheet" TYPE="text/css" HREF="style.css">
<STYLE TYPE="text/css">
<!—
P { text-align: justify; color: green; }
.title { color: blue; font-weight: bold; font-size: 16px; }
-->
</STYLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<FONT CLASS="title">Способы определения шаблонов CSS</FONT>
<P>
Ниже перечислены существующие способы определения стилевых шаблонов CSS,
даны характеристики каждого способа, приведены примеры их использования.
</P>
<UL>
<LI>Ссылка на внешний файл
<LI>Внедрение в документ
<LI>Включение в теговые конструкции
<LI>Импортирование
</UL>
</BODY>
</HTML>
```

Также следует сказать, что использование каждого способа определения стилевых шаблонов CSS может быть связано с некоторыми минусами.

- В случае ошибки интерпретации HTML-кода браузером, плохой связи с сервером и пр. внешний файл CSS может не загрузиться, вследствие чего стиль для нужных элементов HTML не будет переопределен.
- Если внешний файл CSS включает слишком большое количество стилевых шаблонов (что отражается на конечном размере файла), то существует вероятность того, что браузер не сумеет полностью интерпретировать файл CSS или вообще исчерпает лимит времени по загрузке данных. В первом случае стили для части элементов не будут переопределены (браузер успеет "обнаружить" только те правила, которые размещены в верхней части CSS-файла). Во втором случае все элементы страницы останутся без изменения, загрузившись по умолчанию.
- При использовании способа включения стиля в сам документ наличие слишком большого количества шаблонов CSS заметно увеличит конечный размер HTML-страницы, что скажется на времени загрузки документа в браузере.

- При сочетании различных способов определения стилевых шаблонов следует учитывать особенности браузеров. К примеру, Netscape "плохо относится" к использованию символа нижнего подчеркивания (_) в указании классов для элементов HTML (.news_title, ._about и т. д.). Также ряд браузеров (и Netscape в том числе) не подключает стилевой шаблон класса, присвоенного ячейке таблицы <TD>/<TH>: конструкция <TD CLASS="text">Текст</TD> выведет текст по умолчанию, а <TDXFONT CLASS="text">Текст</FONT></TD> отобразит текст заданным стилем.

Запись шаблона CSS

Группировка и наследование

Любое правило таблицы стилей CSS состоит из селектора и определения шаблона.

Селектор выступает в роли указателя стилевого правила для определенного HTML-элемента или внутреннего класса (идентификатора). Определение шаблона — это описание стилевых правил для обозначенных элементов HTML. Правила чередуются через точку с запятой и заключаются в фигурные скобки.

```
H3 { color: blue; font-family: Tahoma, Verdana, Arial; }
```

В данном примере селектором является элемент заголовка нз, для шаблона которого следует такое определение: цвет — синий, шрифт — Tahoma, либо Verdana, либо Arial.

Как видно из примера, для одного селектора приведено описание, содержащее два правила — по цвету заголовка и наименованию гарнитуры. Это говорит о том, что CSS позволяет группировать несколько стилевых правил для одного селектора в рамках единого описания шаблона.

Сравнив запись вида:

```
H3 ( color: blue; }
H3 { font-family: Tahoma, Verdana, Arial; }
```

и

```
H3 { color: blue; font-family: Tahoma, Verdana, Arial; }
```

можно сделать вывод о том, что группировка правил по селектору позволяет, во-первых, экономить размер CSS-файла, во-вторых — систематизировать структуру описания шаблона.

Другой особенностью таблиц стилей CSS является свойство наследования стилевых правил для нескольких селекторов одновременно, например:

```
TD, TH, P, DIV { text-align: justify; color: gray: font-size: 10px; }
```

Такая запись назначает единый стиль отображения текстовой информации для элементов ячейки таблицы (<TD>, <тн>), а также параграфов (<P>)

и блоков (<DIV>), а именно: тип выравнивания — по ширине, цвет — серый, размер шрифта — 10 пикселов.

Селекторы

В качестве селектора CSS могут выступать:

- *элементы HTML*. Переопределение стиля для конкретного элемента страницы:

  ```
  BODY { color: orange; }
  ```

  В этом случае весь текст в пределах раздела BODY будет оранжевым. При добавлении, например, таблицы назначение стилевого шаблона пропадет для текста внутри ячеек;

- *классы*. Использование классов позволяет переопределять стиль как для конкретного элемента, так и для любого элемента, которому присвоен данный класс. Наименование класса начинается с точки и обычно пишется строчными буквами (допускается использование латинских букв и цифр, но наличие специальных символов, нижних подчеркиваний и прочих нестандартных элементов не рекомендуется).

  ```
  .red { color: red; }
  ```

  В этом случае любой элемент HTML, позволяющий менять цвет, будет отображаться красным, если ему присвоить класс .red:

  ```
  <FONT CLASS="red">Текст красным цветом</FONT>
  ```

  или

  ```
  <HR CLASS="red">
  ```

  Если мы дополним селектор класса наименованием конкретного HTML-элемента, то действие стилевого правила будет распространяться только на данный элемент:

  ```
  HR.red { color: red; }
  ```

  При указании классов стилевого шаблона следует внимательно следить за тем, поддерживает ли HTML-элемент присваемый тип переопределения стиля. Например, запись вида:

  ```
  HR { text-align: justify; }
  ```

  будет бессмысленной, т. к. горизонтальный разделитель относится к области структурного форматирования и не может содержать текст, который, согласно стилевому правилу, следует растянуть по ширине;

- *идентификаторы*. Запись идентификатора начинается с символа # (диез) и заканчивается наименованием:

  ```
  #black { background-color: black; }
  ```

  Например, присвоив данный идентификатор тегу TD, мы получим ячейку таблицы, залитую черным цветом:

  ```
  <TD ID="black">Ячейка черного цвета</TD>
  ```

Сравнив функции селектора класса и идентификатора, можно задаться вполне закономерным вопросом — чем же отличаются эти селекторы? Действительно, формат определения селектора обоих типов аналогичен по структуре и присвоению HTML-элементам. Однако селектор идентификатора часто применяется для задания уникального имени элементу, который задействован в программном сценарии (скрипте). В отличие от него, селектор класса ограничивается, в основном, применением в стилевых шаблонах.

В заключение необходимо обратить особое внимание на невозможность сочетания селекторов различных типов. Нельзя одновременно переопределить стиль для стандартного элемента HTML и для него же, но по конкретному классу/идентификатору.

Псевдоклассы

Псевдоклассами называют определенные условия форматирования HTML-элемента, в соответствии с которыми браузер подставляет необходимые стилевые правила отображения данных. При этом в исходной структуре электронного документа такие классы не присутствуют, они создаются в процессе интерпретации HTML-кода браузером.

В основном, псевдоклассы предназначены для задания различных типов форматирования по нескольким разновидностям элементов. Рассмотрим функциональность псевдоклассов на примере гиперссылок.

Согласно спецификации HTML и стандарту CSS, гиперссылка может принимать четыре состояния: непосещенная ссылка (link), посещенная ссылка (visited), активная ссылка (active) и ссылка при наведении курсора мыши (hover). **Первые Три СОСТОЯНИЯ** (link, visited, active) **обычно** прописываются в теге <BODY> HTML-документа (уровень CSS 1). Четвертое состояние (hover) относится к уровню CSS 2 и подразумевает изменение цвета ссылки при наведении на нее курсора мыши пользователя (событие работает в браузерах Internet Explorer 4 и выше, Opera 5 и выше, Netscape 6 и выше, Mozilla 1.0; браузеры более ранних версий, а также некоторые не особенно распространенные браузеры не поддерживают состояние hover).

Эти состояния и будут являться псевдоклассами при записи правил отображения гиперссылок в стилевом шаблоне:

```
A:link { color: blue; }
A:active { text-decoration: underline; }
A:visited { color: gray; }
A:hover { color: orange; }
```

В данном случае все присутствующие в электронном документе гиперссылки будут отображаться в соответствии с заданным стилевым правилом. Однако часто возникает необходимость визуально выделить одни ссылки

относительно других. Для этого наряду с псевдоклассами используются обычные селекторы классов:

```
A:active.red { color: red; }
A:hover.red ( color: blue; }
A:active.white { color: white; }
A:hover.white ( color: black; }
```

Применение таблиц стилей CSS

Прежде всего следует отметить, что при определении стилевых таблиц далеко не всегда свойства стандартного HTML-элемента соответствуют описанию шаблона стиля. Например, в HTML для жирного начертания используется тег-контейнер <в> (<STRONG>), а в CSS — конструкция font-weight: `bold;` (для элемента или селектора). Для выделения текста подчеркиванием в HTML предусмотрен тег `<U>`, а в CSS используется запись вида `text-decoration:` underline; и т.д.

В рамках данной книги, посвященной применению языка разметки HTML, сложно рассказать обо всех свойствах переопределения стиля с помощью CSS. Поэтому мы остановимся лишь на некоторых аспектах использования каскадных таблиц стилей, а именно: на форматировании текста, структурном форматировании и организации пользовательских форм.

CSS в форматировании текста

CSS предоставляет разработчику электронных документов гораздо более широкий набор возможностей работы с текстовой информацией, нежели стандартный HTML. Помимо способов выделения текста (подчеркивание, курсив, жирное начертание, выбор гарнитуры и размер шрифта), с помощью средств CSS возможно изменять такие параметры, как межбуквенный и межстрочный интервал, тип регистра (строчные и прописные буквы) и многое другое.

Единицы измерения в таблицах стилей

Все единицы измерения свойств элементов в CSS можно разделить на абсолютные и относительные (табл. 9.1).

Абсолютные единицы измерения используются в случае, когда известны характеристики того устройства, которое отображает информацию.

Относительные единицы измерения определяют масштаб форматируемого элемента относительно других элементов, что позволяет сохранить первозданность документа при выводе на передающее устройство, характеристики которого заранее не известны.

Таблица 9.1. *Единицы измерения CSS*

Абсолютные	Относительные
in (дюйм ~ 2,5 см)	em (высота шрифта элемента)
mm (миллиметр)	ex (высота буквы x)
cm (сантиметр)	px (пиксел)
pt (пункт ~ 1/7 дюйма)	% (процентное соотношение)
pc (пика = 12 пунктов)	

В табл. 9.2 приведены наиболее распространенные свойства форматирования текста в CSS.

Таблица 9.2. *Свойства форматирования текста в CSS*

Свойство	Формат записи	Функция
`font-family`	`font-family: Tahoma, Arial;`	Выбор гарнитуры для отображения (допускается перечисление нескольких названий через запятую)
`font-size`	`font-size: 11px;`	Размер шрифта
`font-style`	`font-style: italic;`	Выбор наклона текста (курсив)
`font-weight`	`font-weight: bold;`	Наличие/отсутствие жирного начертания
`font-variant`	`font-variant: small-caps;`	Преобразование строчных букв в прописные с изменением пропорций
`text-decoration`	`text-decoration: underline;`	Подчеркивание текста
`text-align`	`text-align: right;`	Определение типа выравнивания текста
`text-transform`	`text-transform: uppercase;`	Выбор регистра букв
`letter-spacing`	`letter-spacing: 1em;`	Межбуквенный интервал
`line-height`	`line-height: 5mm;`	Межстрочный интервал
`Color`	`color: #FFFFFF;`	Цвет текста
`Background-color`	`background-color: white;`	Цвет фона текста

Рассмотрим на примере (рис. 9.1, листинг 9.2) некоторые свойства форматирования текста в CSS.

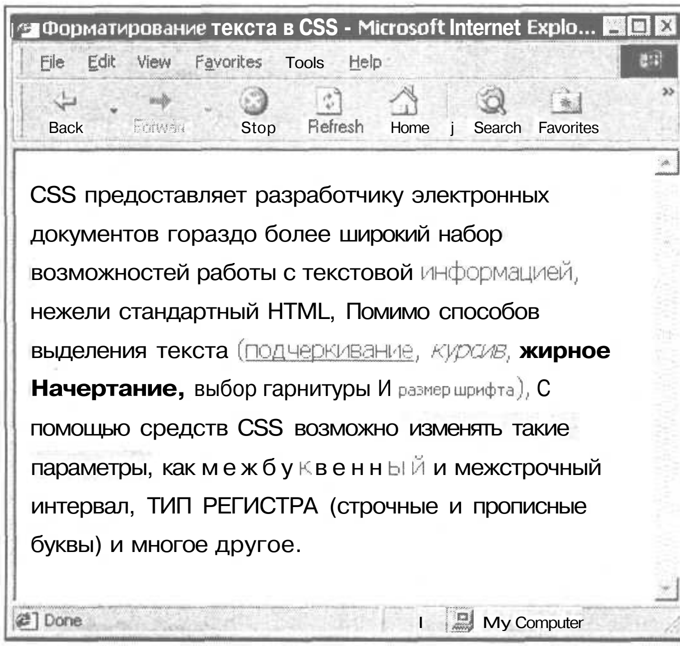


Рис. 9.1. Форматирование текста в CSS

Листинг 9.2. Форматирование текста в CSS

```
<HTML>
  <HEAD>
    <TITLE>Форматирование текста в CSS</TITLE>
<STYLE TYPE="text/css">
<!--
.text { font-family: Tahoma; color: #003366;
        line-height: 7mm; font-size: 12pt; }
#kursiv { font-style: italic; }
SPAN.font { font-size: 11px; }
.color { background-color: #CECECE; }
-->
</STYLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
```

```
<FONT CLASS="text">
CSS предоставляет разработчику электронных документов гораздо более широкий набор возможностей работы с текстовой информацией, нежели стандартный HTML. Помимо способов выделения текста (<FONT STYLE="text-decoration: underline;">подчеркивание</FONT>, <FONT ID="kursiv">курсив</FONT>, <FONT STYLE="font-weight: bold;">жирное начертание</FONT>, <FONT STYLE= "font-family: Times New Roman;">выбор гарнитуры</FONT> и <SPAN CLASS="font">размер шрифта</SPAN>), с помощью средств CSS возможно <FONT CLASS="color">изменять такие параметры</FONT>, как <FONT STYLE= "letter-spacing: 3px;">межбуквенный</FONT> и межстрочный интервал, <FONT STYLE="text-transform: uppercase;">тип регистра</FONT> (строчные и прописные буквы) и многое другое.
</FONT>
</BODY>
</HTML>
```

Структурное форматирование

Также каскадные таблицы стилей CSS привнесли много нового и заметно расширили функциональность форматирования структурных элементов электронного документа: p, DIV и пр.

В табл. 9.3 приведены наиболее распространенные свойства структурного форматирования в CSS.

***Таблица 9.3.** Свойства структурного форматирования в CSS*

Свойство	Формат записи	Функция
border-width	border-width: 20px;	Ширина границы структурного элемента
border-style	border-style: solid;	Тип декоративного отображения границы элемента
border-color	border-color: gray;	Цвет границы структурного элемента
list-style-type	list-style-type: square;	Тип нумерованного или маркированного списка
list-style-image	list-style-image: url(«bullet.gif");	Указание пути к графическому маркеру
margin	margin: 1px 2px 3px 4px;	Определение размера поля относительно верхнего, правого, нижнего и левого края структурного элемента
padding	padding-top: 10em; padding-right: 25px;	Определение отступа от верхнего, правого, нижнего и левого края структурного элемента

***Таблица 9.3** (окончание)*

Свойство	Формат записи	Функция
width	width: 300px;	Ширина структурного элемента
height	height: 120px;	Высота структурного элемента
background-color	background-color: #CECECE;	Цвет фона структурного элемента
float	float: left;	Плавающее расположение структурного элемента относительно других элементов

Рассмотрим на примере (листинг 9.3, рис. 9.2) некоторые свойства структурного форматирования в CSS.

Листинг 9.3. Структурное форматирование в CSS

```
<HTML>
  <HEAD>
    <TITLE>Структурное форматирование в CSS</TITLE>
<STYLE TYPE="text/css">
<!--
.text { font-family: Tahoma; color: #003366;
        line-height: 7mm; font-size: 12pt;
        margin: 10px 10px 10px 10px; padding-left: 15px;
        padding-right: 15px; border-color: black;
        border-style: dotted; background-color: #C0C0C0;
        width: 500px; text-align: justify; }
#kursiv { font-style: italic; }
SPAN.font { font-size: 11px; >
.color { background-color: #CECECE; }
IMG { float: left; width: 100px; height: 100px; }
LI {list-style-type: circle; }
-->
</STYLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
```

```
<P CLASS="text">
<IMG SRC="picture.gif ">CSS предоставляет разработчику электронных
документов гораздо более широкий набор возможностей работы с текстовой
информацией, нежели стандартный HTML. Помимо способов выделения текста:
</P>
<UL>
<LIXFONT STYLE="text-decoration: underline; ">подчеркивание</FONT>
<LIXFONT  ID="kursiv">курсив</FONT>
<LIXFONT STYLE=" font-weight: bold; ">жирное начертание</FONT>
<LIXFONT STYLE="font-family: Times New Roman; ">выбор гарнитуры</FONT>
<LI><SPAN CLASS="font">размер шрифта</SPAN>)
</UL>
<P CLASS="text">
с помощью средств CSS возможно <FONT CLASS="color">изменять такие пара-
метры</FONT>, как <FONT STYLE=" letter-spacing: 3px; ">межбуквенный</FONT>
и межстрочный интервал, <FONT STYLE="text-transform: uppercase; ">тип ре-
гистра</FONT> (строчные и прописные буквы) и многое другое.
</P>
</BODY>
</HTML>
```

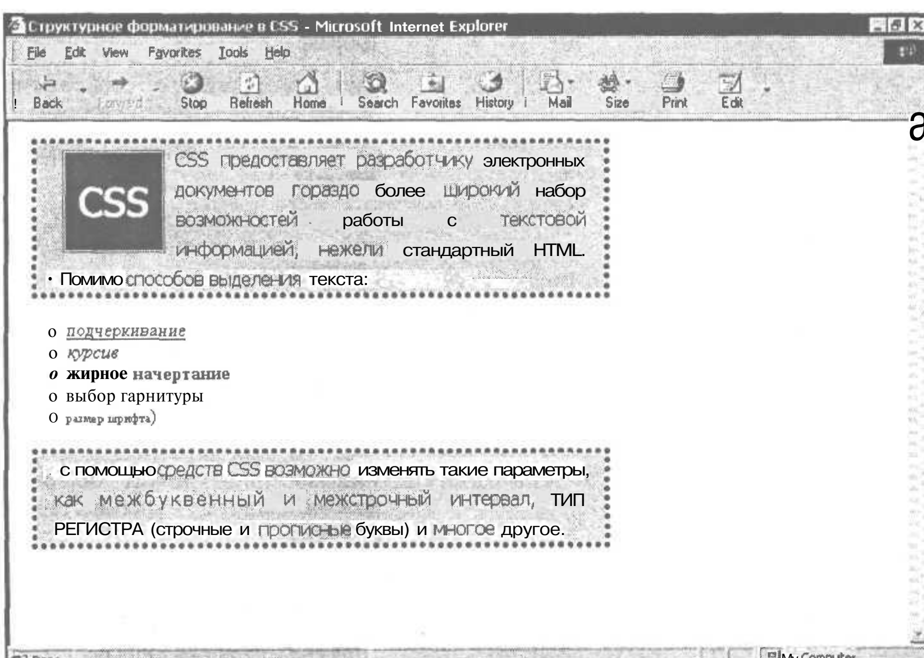


Рис. 9.2. Структурное форматирование в CSS

Как видно из примера, некоторые структурные элементы позволяют применять свойства CSS, относящиеся к форматированию текста: text-align, background-color и др. Таким образом, при формировании таблиц стилей допускается сочетание свойств CSS как для текстового, так и для структурного форматирования.

Пользовательские формы в CSS

Язык разметки HTML предлагает разработчику электронных документов широкий набор возможностей по организации пользовательских форм. Однако при развитой функциональности внешний вид всевозможных текстовых полей, переключателей, ниспадающих меню зачастую оставляет желать лучшего. Это досадное упущение компенсируется посредством таблиц стилей CSS.

В табл. 9.4 приведены наиболее распространенные свойства пользовательских форм в CSS.

Таблица 9.4. *Пользовательские формы в CSS*

Свойство	Формат записи	Функция
Background-color	background-color: white;	Цвет фона элемента формы
border-width	border-width: 1px;	Толщина рамки вокруг элемента формы
border-style	border-style: solid;	Тип декоративного отображения рамки вокруг элемента формы
border-color	border-color: #003366;	Цвет рамки вокруг элемента формы
border-top border-right border-bottom border-left	border-top: 1px; border-right: 2px; border-bottom: 3px; border-left: 4px;	Толщина рамки для верхней, правой, нижней и левой части элемента формы
font-family font-size font-weight	font-family: Tahoma; font-size: 12em; font-weight: bold;	Свойства текста внутри элемента формы (аналогичны свойствам форматирования обычного текста)
margin	margin: 2px 2px 2px 2px	Размер поля вокруг элемента формы (вверху, справа, внизу, слева)
padding	padding-top: 1px; padding-right: 2px;	Размер внутреннего отступа от края элемента формы
width	width: 200px;	Ширина элемента формы
height	height: 50px;	Высота элемента формы

Рассмотрим на примере (листинг 9.4, рис. 9.3) некоторые свойства пользовательских форм в CSS.

Листинг 9.4. Пользовательские формы в CSS

```
<HTML>
  <HEAD>
    <TITLE>Пользовательские формы в CSS</TITLE>
<STYLE TYPE="text/css">
<!—
INPUT.text { font-family: Tahoma; color: #003366;
             font-size: 12px; padding-left: 5px;
             padding-right: 5px; border-color: black;
             border-style: solid; border-width: 1px;
             background-color: #F9F9F9; width: 250px; }

TEXTAREA. field { font-family: Verdana, Arial;
                  color: #003366; font-size: 12px;
                  padding-left: 5px; padding-right: 5px;
                  border-color: black; border-style: solid;
                  border-width: 1px; background-color: #F9F9F9; }

INPUT.button { font-family: Tahoma; color: #FFCC00;
               font-size: 11px; font-weight: bold;
               border-color: white; border-style: solid;
               border-width: 1px; background-color: gray;
               width: 200px; }

BODY, TD { font-family: Tahoma; color: #003366;
           font-size: 11px; }

TH { color: #FFFFFF; }
-->
</STYLE>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<H2 ALIGN="center">ПОЛЬЗОВАТЕЛЬСКАЯ АНКЕТА</H2>
<FORM NAME="user_anketa" ACTION="/cgi-bin/anketa.cgi" METHOD="post"
ENCTYPE="multipart/form-data">
```

```
<TABLE BORDER="0" CELLPADDING="5" CELLSPACING="3" ALIGN="center"
WIDTH="98%">
<TR>
<TH BGCOLOR="#999999" WIDTH="50%">Личные данные</TH>
<TH BGCOLOR="#999999" WIDTH="50%">Увлечения</TH>
</TR>
<TR>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
Ваше имя:<BR>
<INPUT TYPE="text" SIZE="30" MAXLENGTH="35" NAME="name" CLASS="text">
<BR>
Ваш пароль:<BR>
<INPUT TYPE="password" SIZE="30" MAXLENGTH="35" NAME="password"
CLASS="text">
<BR>
Ваша фотография:<BR>
<INPUT TYPE="file" SIZE="18" NAME="photo" CLASS="text">
</TD>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
<INPUT TYPE="checkbox" NAME="hobby" VALUE="computers" CHECKED>Компьютеры
и Интернет
<BR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="art">Литература и искусство
<BR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="music">Музыка и танцы
<BR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="auto">Автомобили
<BR>
<INPUT TYPE="checkbox" NAME="hobby" VALUE="sport">Спорт и активный отдых
</TD>
</TR>
<TR>
<TH BGCOLOR="#999999" WIDTH="50%">Место жительства</TH>
<TH BGCOLOR="#999999" WIDTH="50%">Комментарии</TH>
</TR>
<TR>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
<SELECT NAME="city" SIZE="5" STYLE="font-family: Verdana; font-size:
11px;">
 <OPTION VALUE="" STYLE="background-color: #CECECE;">--- Выберите город ---
```

```
 <OPTION VALUE="Moscow">Москва
 <OPTION VALUE="Spb" STYLE="background-color: #CECECE;">Санкт-Петербург
 <OPTION VALUE="Samara">Самара
 <OPTION VALUE="Ryazan" STYLE="background-color: #CECECE;">Рязань
 <OPTION VALUE="Krasnoyarsk">Красноярск
 <OPTION VALUE="Komsomolsk" STYLE="background-color:
ICECECE; ">Комсомольск-на-Амуре
 <OPTION VALUE="Vladivostok">Владивосток
</SELECT>
<BRXBR>
<SELECT STYLE="font-family: Verdana; font-size: 11px;">
 <OPTION VALUE="">——Выберите страну ---
 <OPTION VALUE="Russia">Россия
 <OPTION VALUE="England">Англия
 <OPTION VALUE="USA">США
 <OPTION VALUE="Germany">Германия
 <OPTION VALUE="France">Франция
</SELECT>
</TD>
<TD VALIGN="top" BGCOLOR="#EEEEEE" WIDTH="50%">
<TEXTAREA COLS="25" ROWS="5" NAME="comments" CLASS="field"></TEXTAREA>
<BR>
<INPUT TYPE="radio" NAME="site" VALUE="first" CHECKED>Я впервые на этом
сайте
<BR>
<INPUT TYPE="radio" NAME="site" VALUE="not_first">Я тут не в первый раз
</TD>
</TR>
<TR>
<TD COLSPAN="2" ALIGN="center" BGCOLOR="#999900">
<INPUT TYPE="submit" VALUE="Отправить" CLASS="button"> <INPUT
TYPE="reset" VALUE="Очистить" CLASS="button">
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

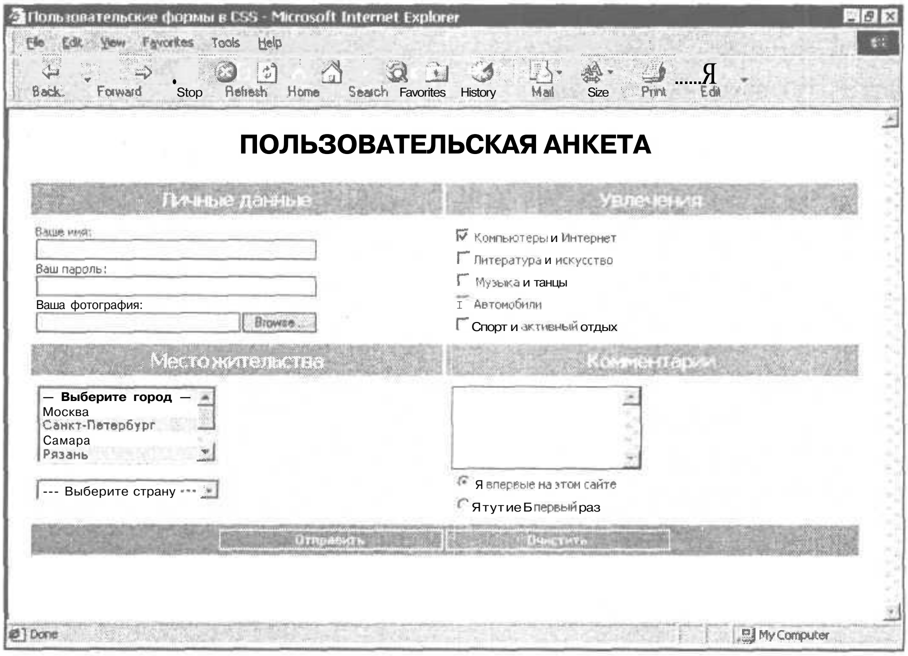


Рис. 9.3. Пользовательские формы в CSS

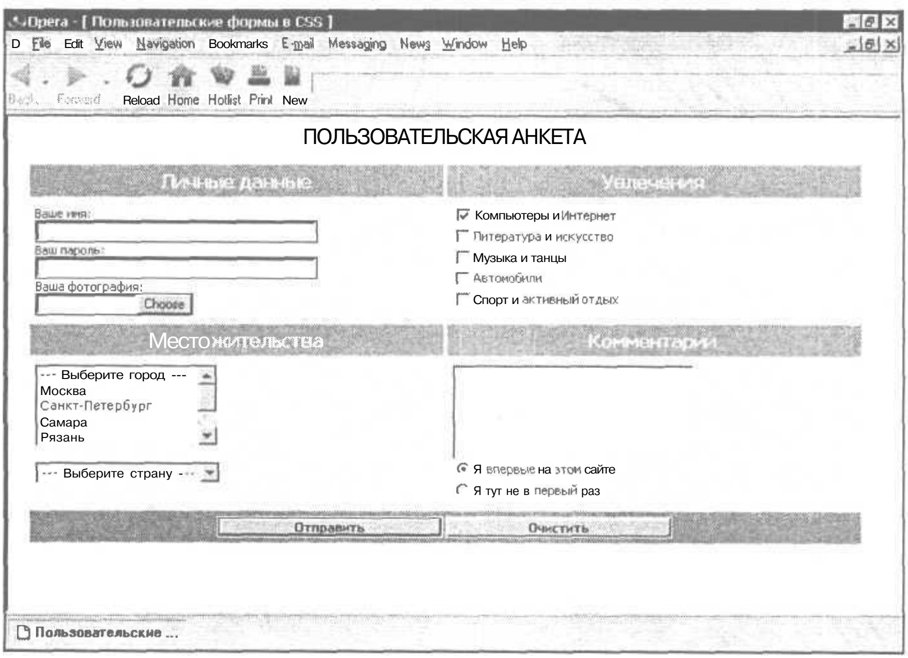


Рис. 9.4. Пользовательские формы в браузере Opera 6 (реализация с CSS)

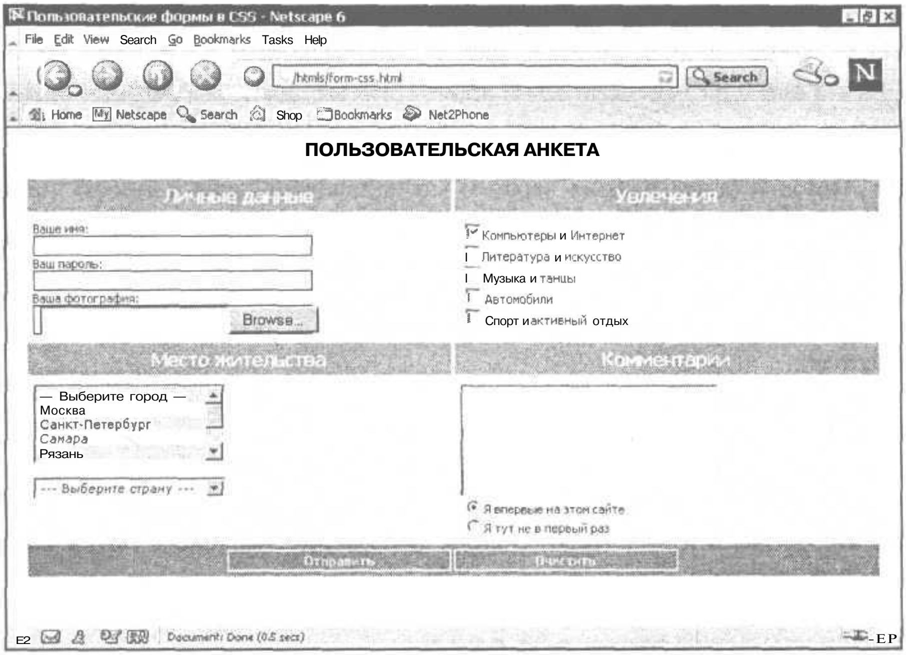


Рис. 9.5. Пользовательские формы в браузере Netscape *6.2* (реализация с CSS)

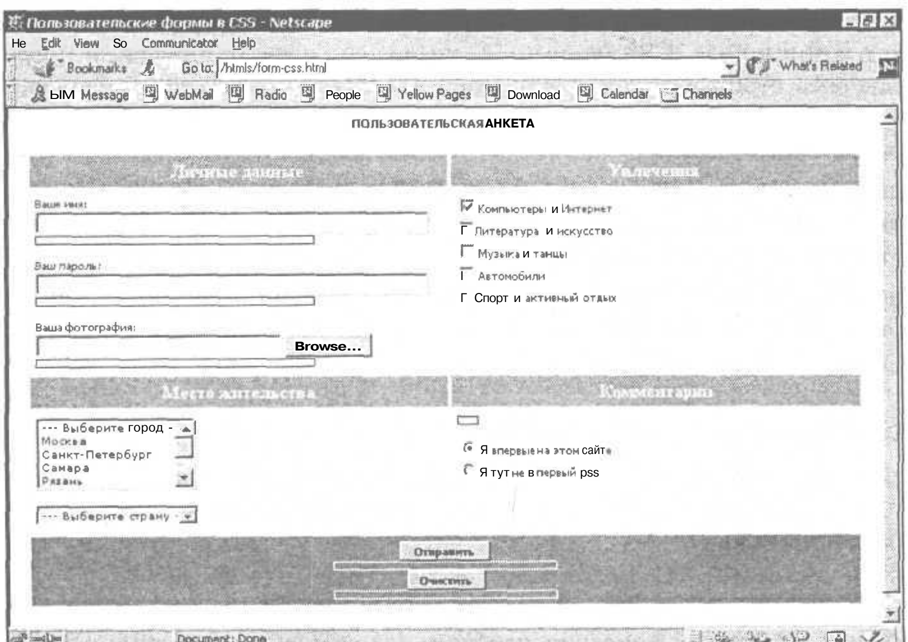


Рис. 9.6. Пользовательские формы в браузере Netscape Navigator 4.7 (реализация с CSS)

Чтобы понять превосходство стилевых шаблонов в организации пользовательских форм, просто сравните рис. 9.3 (применение таблиц стилей CSS) и рис. 8.16 (стандартные элементы форм HTML).

Однако и здесь есть подводные камни. Если, к примеру, Internet Explorer поддерживает практически все свойства CSS 1 и CSS 2, то другие браузеры (в особенности более ранних версий) могут искажать или попросту не отображать стилевые шаблоны (рис. 9.4).

Браузер Opera 6 не отображает цвет фона для элементов списка SELECT, не поддерживает свойство border-style в текстовых полях INPUT, свойство color в наименовании кнопок подтверждения (Отправить) и отмены (Очистить).

Браузер Netscape 6.2 (рис. 9.5) более лоялен к отображению свойств стилевых шаблонов CSS, однако некорректно интерпретирует конструкцию INPUT TYPE="file" (загрузка файла с локального компьютера).

В отличие от своей более усовершенствованной версии, Netscape Navigator 4.7 выводит на экран полное нарушение стилей, а также компоновки электронного документа (рис. 9.6).

Позиционирование объектов

Исходя из концепции языка разметки HTML, все элементы документа выводятся браузером в той последовательности, в какой размещены теговые конструкции в коде. CSS (уровень 2) позволяет задавать порядок и последовательность отображения тех или иных HTML-элементов в зависимости от определенных событий на странице или манипуляций, осуществляемых со стороны пользователя.

Другими словами, с помощью средств CSS можно прибегнуть к позиционированию (пространственному расположению) необходимых объектов в пределах электронного документа.

Существуют два типа визуального позиционирования элементов: абсолютное и относительное. Для описания каждого типа используются специальные свойства position, top И left.

Абсолютное позиционирование

Абсолютное позиционирование подразумевает четкое фиксирование выбранного элемента на странице, независимо от остальных элементов документа, например:

```
<IMG SRC="picture.gif" WIDTH="100" HEIGHT="100" ALT="Рисунок"
STYLE="position: absolute; top: 10px; left: 25px;">
```

В данном случае графическое изображение абсолютно спозиционировано и размещается в 10 пикселах от верхнего и в 25 пикселах от левого края

своего родительского элемента (в качестве родительского элемента выступает верхняя левая точка структуры документа).

Относительное позиционирование

Относительное позиционирование позволяет расположить выбранный объект в зависимости от размещения остальных элементов документа (т. е. относительно других объектов страницы), например:

```
<DIV ID="text" STYLE="position: relative; top: 50px; left: 50px;">
```

Dynamic HTML

Dynamic HTML или динамический HTML не является отдельным языком программирования или разметки документа. Это всего лишь технология, реализующая электронные документы с динамически изменяющимся содержанием.

Реализация динамического HTML строится на трех компонентах:

- HTML — HyperText Markup Language. Простой язык разметки гипертекстовых документов;
- CSS — Cascading Style Sheets. Каскадные таблицы стилей HTML-документа;
- JavaScript — клиентский язык программирования (выполняется не на сервере, а непосредственно в браузере пользователя на его локальном компьютере). Возможный аналог — VBScript, JScript.

Эти три компонента образуют важнейшую структуру под названием Document Object Model, DOM (Объектная модель документа), которая в дополнение к простоте HTML и изяществу CSS обладает свойством возможности динамического изменения содержания без перезагрузки электронного документа.

К сожалению, объектная модель DHTML (Dynamic HTML), входящая в состав Internet Explorer (начиная с версии 4.01), по многим параметрам не поддерживается браузерами других производителей (например, Opera или Netscape), что накладывает определенные ограничения на процесс разработки интерактивных документов с динамическим содержанием.

Создание визуальных эффектов

Динамический HTML позволяет разработчику создать в пределах электронного документа рабочий инструментарий, позволяющий пользователю манипулировать содержанием страницы, видом и расположением объектов или элементов и т. д.

Рассмотрим действие Dynamic HTML на примере динамических информационных блоков и применения визуальных фильтров для графических изображений.

Динамические блоки

Область применения динамических информационных блоков чрезвычайно широка, поэтому остановимся лишь на одном, достаточно распространенном случае.

Предположим, что на сайте расположена карта России с нанесенными на ней названиями городов. Посетитель должен быстро и в удобном виде получить информацию по каждому из городов.

Какие могут быть способы реализации данной задачи? Разрезать карту на части или задать активные области-изображения (Imagemap) для нее? Неудобно, потому что каждый раз посетитель будет переходить по ссылке и назад, что снова потребует загрузки графического файла.

Поместить нужный текст в поле ALT, чтобы тот отображался в виде всплывающей подсказки при наведении курсора? Бессмысленно, поскольку подсказка через несколько секунд исчезнет, а текст может быть большим (десятки предложений и т. п.).

Самый подходящий и эффективный способ — прибегнуть к помощи Dynamic HTML в совокупности с Imagemap. В этом случае при наведении курсора мыши на заданную активную область карты в определенном месте страницы появится информация о городе (как текстовая, так и графическая). Такой подход имеет два достоинства — корректная работа (что в Internet Explorer, что в Netscape Navigator), да еще экономия времени посетителя (графика загружается только один раз, а все действия выполняются непосредственно из кода текущего электронного документа).

Но для этого прежде всего необходимо определить JavaScript-сценарий в разделе HEAD (листинг 9.5):

Листинг 9.5. JavaScript-сценарий для динамических информационных блоков

```
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
var ie = document.all ? 1 : 0;
var ns = document.layers ? 1 : 0;
var topcss = 165;
if (ns)
topcss = 200;
function showLayer(name)
{
```

```
if (ie)
document.all[name].style.visibility = "visible";
else if (ns)
document.layers[name].visibility = "show";
}
function hideLayer(name)
(
if (ie)
document.all[name].style.visibility = "hidden";
else if (ns)
document.layers[name].visibility = "hide";
}
//-->
</SCRIPT>
```

Далее определяем координаты информационных слоев (листинг 9.6), которые невидимы до тех пор, пока курсор мыши не переместится на активную область (координаты размещаются в начале раздела BODY):

Листинг 9.6. Координаты информационных слоев

```
<STYLE TYPE="text/css">
<! —
#linkl
    { position: absolute; left: 160px; top: 250px; visibility: hidden; }
#link2
    { position: absolute; left: 160px; top: 250px; visibility: hidden; }
ilink3
    { position: absolute; left: 160px; top: 250px; visibility: hidden; )
#link4
    { position: absolute; left: 160px; top: 250px; visibility: hidden; }
//-->
</STYLE>
```

В данном случае в качестве селектора стилевого шаблона используется идентификатор.

Поле

```
#link3
    {position: absolute; left: 160px; top: 250px; visibility: hidden; )
```

означает, что при активации области link3 скрытый до этого момента информационный слой, соответствующий данной области, появится на странице на 160 пикселов левее границы окна и на 250 пикселов ниже его верхнего края.

Далее переходим к наполнению самих информационных слоев, которые лучше всего реализовать с помощью структурного тега <DIV> (листинг 9.7):

Листинг 9.7. Наполнение информационных слоев

```
<DIV ID='link1'>Санкт-Петербург</DIV>
<DIV ID='link2'>Ростов-на-Дону</DIV>
<DIV ID='link3'>Красноярск</DIV>
<DIV ID='link4'>Владивосток</DIV>
```

Внутри тега-контейнера <DIV> можно разместить таблицы, графические объекты, нумерованные и маркированные списки и пр.

Наконец, последнее, что нужно сделать, — это задать активные области при помощи Imagemap и привязать их к функциям сценария по активации/дезактивации слоев (листинг 9.8):

Листинг 9.8. Определение активных областей Imagemap и привязка к JavaScript-сценарию

```
<IMG SRC="russiamap.gif" WIDTH="400" HEIGHT="230" BORDER="0"
USEMAP="#russia">

<MAP NAME="russia">

<AREA SHAPE="rect" COORDS="63,70,162,85" HREF="link1.html"
onMouseOver="showLayer('link1');" onMouseOut="hideLayer('link1');">

<AREA SHAPE="rect" COORDS="98,137,193,157" HREF="link2.html"
onMouseOver="showLayer('link2');" onMouseOut="hideLayer('link2');">

<AREA SHAPE="rect" COORDS="249,125,326,141" HREF="link3.html"
onMouseOver="showLayer('link3');" onMouseOut="hideLayer('link3');">

<AREA SHAPE="rect" COORDS="289,89,368,105" HREF="link4.html"
onMouseOver="showLayer('link4');" onMouseOut="hideLayer('link4');">

</MAP>
```

Конструкции onMouseOver и onMouseOut являются событиями языка JavaScript (наведение и снятие курсора мыши с активной области на карте). Каждой активной области с заданными координатами должен соответствовать информационный блок со своим идентификатором (link1, link2, link3 и т.д.).

В результате при перемещении курсора, например, над надписью "Санкт-Петербург" на карте аналогичная надпись, но уже в текстовом виде, появляется прямо под изображением карты России. Как уже было сказано, наполнение слоев можно осуществлять в соответствии с индивидуальными предпочтениями

и оформительскими требованиями (вместо текстовых блоков вставлять графику, таблицы, гиперссылки и др.). В листинге 9.9 и, соответственно, на рис. 9.7 приведен пример применения динамических информационных блоков.



Рис. 9.7. Применение динамических информационных блоков (в примере использованы некоторые стили CSS)

Листинг 9.9. Применение динамических информационных блоков (итоговый листинг)

```
<HTML>
  <HEAD>
    <TITLE>Применение динамических информационных блоков</TITLE>
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
var ie = document.all ? 1 : 0;
var ns = document.layers ? 1 : 0;
var topcss = 165;
if (ns)
topcss = 200;
function showLayer(name)
```

```
{
if (ie)
document.all[name].style.visibility= "visible";
else if (ns)
document.layers[name].visibility= "show";
}
function hideLayer (name)
{
if (ie)
document.all[name].style.visibility = "hidden";
else if (ns)
document.layers[name].visibility= "hide";
}
//-->
</SCRIPT>
  </HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="black" LINK="#00FF00" ALINK="#00FF00"
VLINK="blue">
<STYLE TYPE="text/css">
<!--
#link1
      ( position: absolute; left: 160px; top: 250px; visibility: hidden; }
#link2
      { position: absolute; left: 160px; top: 250px; visibility: hidden; }
#link3
      { position: absolute; left: 160px; top: 250px; visibility: hidden; }
#link4
      ( position: absolute; left: 160px; top: 250px; visibility: hidden; }
//-->
</STYLE>
<DIV ID='link1'>Санкт-Петербург</DIV>
<DIV ID='link2'>Ростов-на-Дону</DIV>
<DIV ID='link3'>Красноярск</DIV>
<DIV ID='link4'>Владивосток</DIV>
<IMG SRC="russiamap.gif" WIDTH="400" HEIGHT="230" BORDER="0"
USEMAP="#russia">
<MAP NAME="russia">
<AREA SHAPE="rect" COORDS="63,70,162,85" HREF="link1.html"
onMouseOver="showLayer('link1');" onMouseOut="hideLayer('link1');">
```

```
<AREA SHAPE="rect" COORDS="98,137,193,157" HREF="link2.html"
onMouseOver="showLayer('link2');" onMouseOut="hideLayer('link2');">
<AREA SHAPE="rect" COORDS="249,125,326,141" HREF="link3.html"
onMouseOver="showLayer('link3');" onMouseOut="hideLayer('link3');">
<AREA SHAPE="rect" COORDS="289,89,368,105" HREF="link4.html"
onMouseOver="showLayer('link4');" onMouseOut="hideLayer('link4');">
</MAP>
</BODY>
</HTML>
```

Визуальныефильтры

В Dynamic HTML под фильтром принято понимать некую функцию, так или иначе преобразующую внешний вид элемента на Web-странице. Преобразование осуществляется непосредственно в браузере, т. е. на стороне клиента (подключение сервера не требуется).

Следует отметить, что применение визуальных фильтров возможно только в браузерах Internet Explorer *4.x* и выше (последняя версия Netscape, по утверждению разработчиков, способна поддерживать только лишь часть функциональных возможностей DHTML). Тем не менее, горячие поклонники Microsoft Internet Explorer или разработчики, ориентирующиеся при создании Web-сайтов на показатели статистики в отношении применяемых браузеров, с помощью средств динамического HTML смогут существенно улучшить внешний вид своих электронных данных.

Прежде всего визуальные динамические фильтры можно применить не ко всем элементам HTML-документа, а только к тем, которые способны определить в браузере блок прямоугольного вида, но при этом сами не являются окнами (например, "плавающие" фреймы).

Таблица 9.5. *Элементы визуальных динамических фильтров*

Наименование элемента	Режим применения
BODY	Всегда
BUTTON	Всегда
DIV	При заданных параметрах ширины и высоты или при использовании абсолютного позиционирования
IMG	Всегда
INPUT	Всегда
SPAN	При заданных параметрах ширины и высоты или при использовании абсолютного позиционирования

Таблица 9.5 *(окончание)*

Наименование элемента	Режим применения
TABLE	Всегда
TD, TH	Всегда
TEXTAREA	Всегда

Формат записи фильтра достаточно прост и аналогичен правилам задания свойств элементов с помощью тега <STYLE>. Запись производится в следующем виде:

```
filter: название_фильтра(параметры)
```

где параметры определяются в стандартном для HTML виде:

```
название_параметра=значение_параметра
```

Прежде чем перейти к рассмотрению фильтров и возможностей их применения, необходимо сказать, что, во-первых, допускается использование сразу нескольких фильтров (если это не противоречит окончательному визуальному результату отображения элемента), во-вторых — при указании фильтров, не имеющих никаких параметров, присутствие круглых скобок (без пробелов) после названия фильтра обязательно.

Общие свойства и описание некоторых фильтров

При использовании фильтров следует помнить, что некоторые из них имеют общие свойства, влияющие на характер действия заданных параметров фильтра. К таковым относятся enabled (со значением true или false, соответственно разрешающим или запрещающим применение фильтра, присоединенного к элементу документа), direction (определяет направление действия таких фильтров, как shadow, blur и др.), strength (задает интенсивность действия фильтра со значением от 0 до 255) и т. п. Часть фильтров помимо общих свойств имеют различные методы их определения (например, фильтр light).

Характеристика свойств и методов некоторых визуальных фильтров приведена в табл. 9.6.

***Таблица 9.6.** Характеристика свойств и методов некоторых визуальных фильтров*

Название фильтра	Свойства	Методы
Blur	add, direction, enabled, strength	–
Chroma	color, enabled	–
Glow	color, enabled, strength	–

Таблица 9.6 (окончание)

Название фильтра	Свойства	Методы
Invert	Enabled	-
Light	Enabled	addAmbient, addCone, addPoint, moveLight и др.
Redirect	Enabled	elementImage
Shadow	color, direction, enabled	-
Xray	Enabled	-

Браузер Internet Explorer 4.x (и выше) поддерживает достаточно большое количество фильтров, широкие возможности по настройке которых представляют применение этой технологии в более выгодном свете.

Таблица 9.7. *Характеристика визуальных фильтров, работающих в браузере Internet Explorer*

Название фильтра	Описание действия
alpha	Определение степени прозрачности объекта
blendTrans	Настройка контрастности отображения объекта
blur	Размытие объекта
chroma	Установление прозрачности пикселам заданного цвета
dropShadow	Создание сплошного силуэта объекта
flipH	Зеркальное отображение объекта по горизонтали
flipV	Зеркальное отображение объекта по вертикали
glow	Создание эффекта свечения внешних границ объекта
gray	Отображение объекта в серых тонах
invert	Инвертирование цветовой гаммы объекта
light	Создание эффекта освещения объекта
mask	Создание прозрачной маски из непрозрачных пикселов объекта
redirect	Преобразование объекта в элемент DAImage (технология Microsoft DirectAnimation)
revealTrans	Эффект появления/исчезновения объекта
shadow	Создание силуэта объекта вдоль его границы
wave	Эффект искривления объекта по вертикали
xray	Изменение глубины цвета объекта (эффект рентгеновского снимка)

Рассмотрим на примере возможные варианты действия некоторых фильтров.

Фильтр *WAVE*

Фильтр wave создает синусоидальное искривление объекта в вертикальном **направлении И имеет свойства** add, enabled, freq, lightStrength, phase Иstrength.

Свойства enabled и strength были описаны ранее, а остальные имеют следующее назначение:

- add — определяет необходимость добавления исходного вида объекта в его отфильтрованную интерпретацию;
- freq — задает количество максимумов в волне искривления объекта;
- lightStrength — добавляет эффект трехмерности гребням волны искажения объекта;
- phase — определяет фазу смещения волны (ее значение задается в процентах относительно начальной фазы, равной 0).

В листинге 9.10 и соответственно на рис. 9.8 показано три вида объекта:

- исходный объект без действия фильтра wave;
- объект с действием фильтра wave и заданными свойствами strength=5, add=0, lightStrength=20;
- **Объект С заданными свойствами** strength=5, add=0, phase=50, lightStrength=20, freq=30.

Листинг 9.10. Использование визуального фильтра wave

```
<!-- Рисунок 1 -->
<IMG SRC="flower.jpg" WIDTH="243" HEIGHT="262">
<!-- Рисунок 2 -->
<IMG SRC="flower.jpg" WIDTH="243" HEIGHT="262" STYLE="filter:
wave(strength=5, add=0, lightstrength=20)">
<!-- Рисунок 3 -->
<IMG SRC="flower.jpg" WIDTH="243" HEIGHT="262" STYLE="filter:
wave(strength=5, add=0, phase=50, lightstrength=20, freq=30)">
```

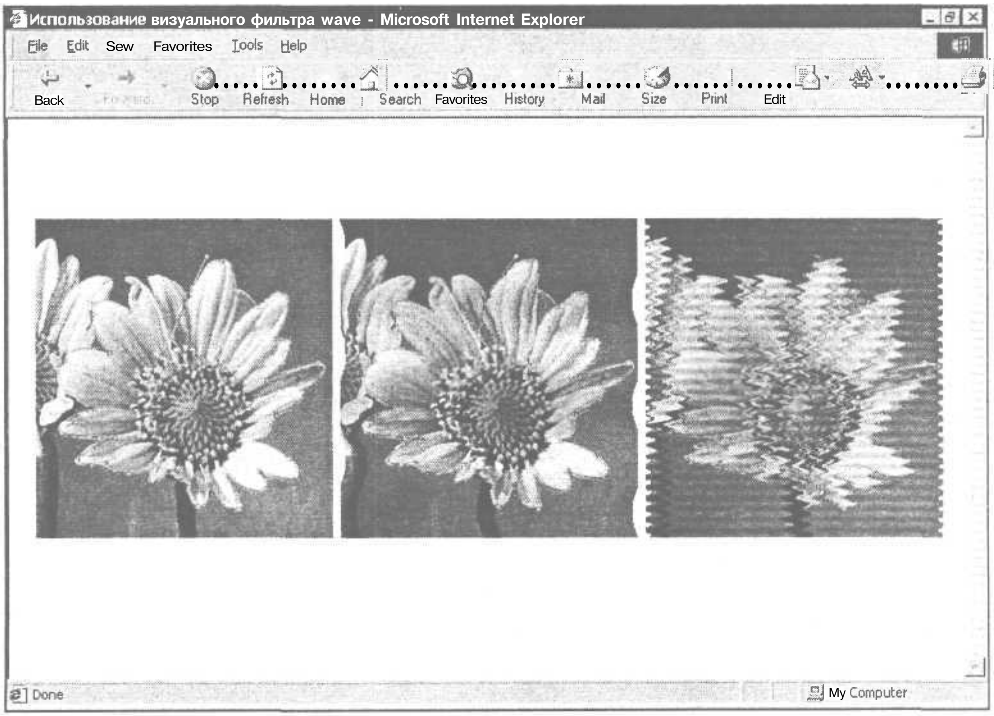


Рис. 9.8. Использование визуального фильтра wave

Фильтр *ALPHA*

Данный фильтр использует такие свойства, как opacity, style и finishopacity. Свойство opacity позволяет определить степень прозрачности объекта. Свойство style может принимать значения от 0 до 3 (нулевое значение соответствует степени прозрачности объекта согласно значению свойства opacity).

Конструкция `style=1` задает координаты точек прямой, на которой степень прозрачности соответствует значению свойства `finishOpacity`.

При style=2, степень прозрачности изменяется от центра объекта к вымышленной окружности, спроецированной на блок отображения объекта (заданный отрезок от значения opacity до значения finishopacity). Если свойство `style=3`, то прозрачность меняется от центра объекта к его фаницам.

На рис. 9.9 представлены три примера действия фильтра alpha:

- `style=0, opacity=60`
- `style=3, opacity=90, finishOpacity=0`
- `style=2, opacity=90, finishOpacity=0`

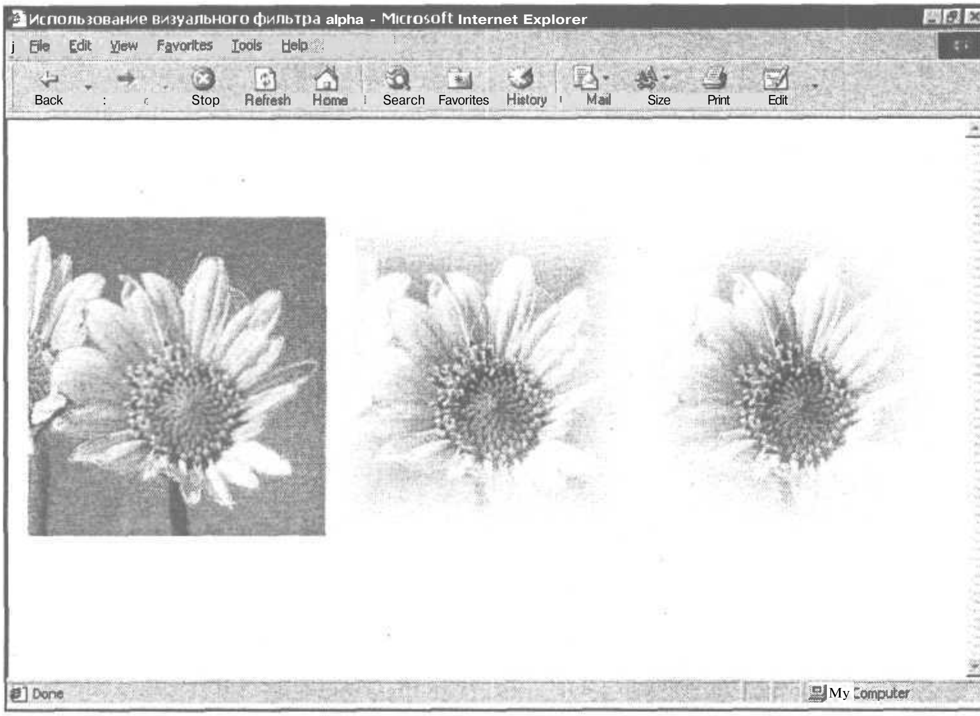


Рис. 9.9. Использование визуального фильтра alpha

Фильтр *dropShadow*

Фильтр dropShadow создает сплошной силуэт объекта, смещенный в заданном направлении, и образует эффект тени.

Кроме enabled, фильтр имеет следующие свойства: color (определение цвета тени), positive (выбор между тенью от прозрачных или непрозрачных точек объекта), offX и offY (расстояние в пикселах в вертикальной и горизонтальной плоскостях, на которых будет отображаться тень от исходного объекта). Положительные значения offx и offY смещают тень, соответственно, вправо и вниз, отрицательные — влево и вверх.

Фильтр *LIGHT*

Данный фильтр имеет единственное свойство enabled и несколько интересных методов, позволяющих варьировать способы освещения выбранного объекта:

- метод addAmbient помещает над объектом источник рассеянного света и требует задания значений по шкале RGB и степени интенсивности наложения iStrength (ОТ О ДО 255);

- метод addCone направляет на объект конический источник света и имеет следующие значения: координаты источника света iX1 и iY1, номер слоя iZ, координаты точки направления света iX2 и iY2, цветовая гамма по шкале RGB (iRed, iGreen, iBlue), интенсивность источника света iStrength и величина угла конуса источника света iAngle;
- метод addPoint имеет те же значения, что и addcone (кроме iX2 и iY2), и добавляет к объекту точечный источник света.

Это наиболее популярные методы, визуализацию двух первых из которых можно увидеть на рис. 9.10.

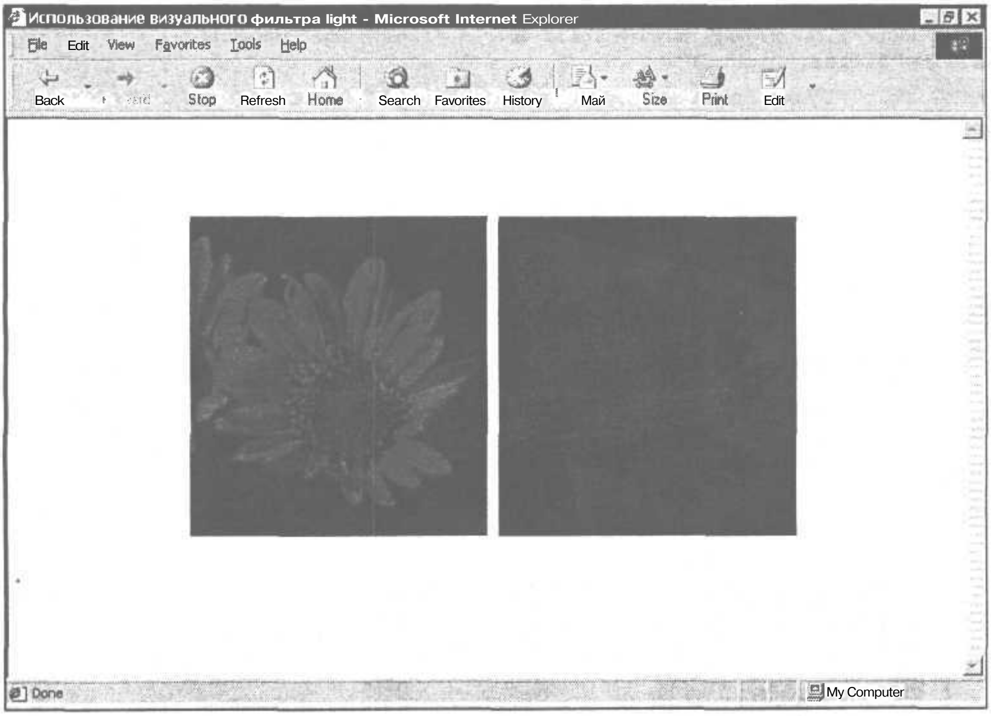


Рис. 9.10. Использование визуального фильтра **light**

Для реализации фильтра light в код HTML-документа обычно включают небольшой JavaScript-сценарий (раздел HEAD) (листинг 9.11):

Листинг 9.11. Использование визуального фильтра light

```
<SCRIPT LANGUAGE="JavaScript">
function lightFilter()
{
```

```
ambient.filters.light.addAmbient(255,0,0,255);
cone.filters.light.addCone(0,0,3,255,255,0,0,255,200,30);
}
</SCRIPT>
```

В основной HTML-код добавляем идентификаторы графических изображений ambient и cone и указываем фильтр light, а также инициализируем функцию фильтра lightFilter() посредством события onLoad в разделе BODY:

```
<BODY onLoad="lightFilter();">
<IMG SRC="flower.jpg" WIDTH="243" HEIGHT="262" ID="ambient"
STYLE="filter: light()">
<IMG SRC="flower.jpg" WIDTH="243" HEIGHT="262" ID="cone" STYLE="filter:
light()">
</BODY>
```

Совместное применение нескольких фильтров

Динамический **HTML** позволяет использовать для одного объекта одновременно несколько фильтров. В этом случае запись фильтров идет через пробел с указанием параметров для каждого из применяемых фильтров.

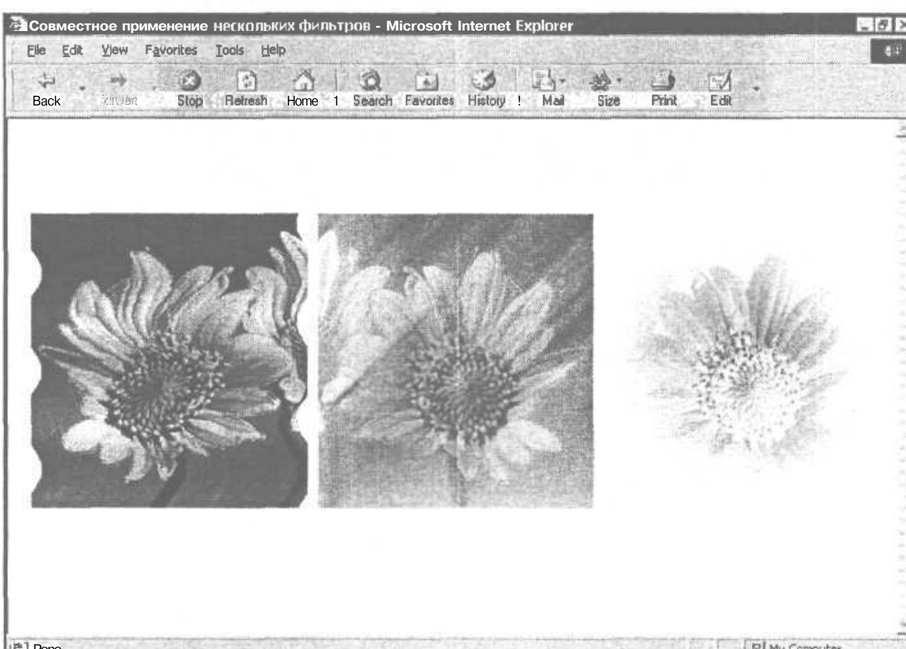


Рис. 9.11. Совместное применение нескольких фильтров

На рис. 9.11 приведены примеры совместного использования нескольких фильтров:

- wave(strength=10, add=0, phase=15, lightStrength=25) flipH()

  Фильтр wave без включения исходного объекта, с интенсивностью наложения 10, фазой смещения 15, трехмерным освещением, равным 25, и фильтр flipH, переворачивающий объект в горизонтальной плоскости;

- gray() blur(strength=85, add=1, direction=45)

  Фильтр gray, убирающий цветовую гамму объекта, и фильтр blur, размывающий изображение, с интенсивностью наложения 85, добавлением исходного объекта и с направлением размытия в 45°;

- xray() alpha(opacity=255, style=2, finishOpacity=0)

  Фильтр xray, создающий эффект рентгеновского снимка, и фильтр alpha со степенью прозрачности объекта 255 и свойством стиля 2 (от центра к воображаемой окружности).

Заключение

В этой книге был рассмотрен язык разметки гипертекстовых документов HTML, его основные функции, свойства и параметры. Сегодня применение HTML практикуется во всех без исключения электронных документах, независимо от тематики, величины и коммерческой направленности интернет-проекта.

Однако мир меняется и современные информационные технологии тоже не стоят на месте: применение стандартных функций HTML становится недостаточным для Web-разработчика, который стремится к максимальной интерактивности на страницах своего сайта.

В данной книге такие технологии, как CSS, Dynamic HTML, JavaScript, CGI, были затронуты лишь поверхностно, дабы показать эффективность использования HTML в совокупности с интерактивными скриптовыми технологиями. Использование последних является отдельной темой, подходящей для отдельной книги.

Что касается HTML, то можно с уверенностью утверждать, что, прочитав эту книгу, вы освоили азы использования данного языка разметки и при желании способны развивать свои навыки в области разработки электронных документов.

Желаю вам удачи в ваших начинаниях!

Приложение 1

Перечень специальных символов HTML

Таблица П1.1

Символ	Код	Вид	Описание
			неразрывный пробел
¡	¡	¡	перевернутый восклицательный знак
¢	¢	¢	цент
£	£	£	фунт стерлингов
¤	¤	a	денежная единица
¥	¥	¥	иена или юань
¦	¦	|	разорванная вертикальная черта
§	§	§	параграф
¨	¨	¨	трема (знак над гласной в иностранных алфавитах)
©	©	©	знак copyright
ª	ª	ª	женский порядковый числитель
«	«	к	левая двойная угловая скобка
¬	¬	¬	знак отрицания
		-	место возможного переноса
®	®	®	знак зарегистрированной торговой марки
¯	¯	¯	знак долготы над гласным
°	°	°	градус
±	±	±	плюс-минус
²	²	^{2}	верхний индекс 'два' - "в квадрате"
³	³	^{3}	верхний индекс 'три' - "в кубе"
´	´	´	знак ударения

Таблица П1.1 (продолжение)

Символ	Код	Вид	Описание
µ	µ	µ	Микро
¶	¶	¶	символ параграфа
·	·	·	точка
¸	¸	-	седиль (орфографический знак в иностранных алфавитах)
¹	¹	1	верхний индекс 'один'
º	º	º	мужской порядковый числитель
»	»	»	правая двойная угловая скобка
¼	¼	¼	дробь - одна четверть
½	½	½	дробь - одна вторая
¾	¾	¾	дробь - три четверти
¿	¿	ȯ	перевернутый вопросительный знак
À	À	A	латинская заглавная буква А с тупым ударением
Á	Á	A	латинская заглавная буква А с острым ударением
Â	Â	A	латинская заглавная буква А с циркумфлексом (диакритический знак над гласной)
Ã	Ã	Ã	латинская заглавная буква А с тильдой
Ä	Ä	Ä	латинская заглавная буква А с тремой
Å	Å	A	латинская заглавная буква А с верхним кружком
Æ	Æ	Æ	латинские заглавные символы АЕ
Ç	Ç	C	латинская заглавная буква С с седилем
È	È	Ë	латинская заглавная буква Е с тупым ударением
É	É	Ë	латинская заглавная буква Е с острым ударением
Ê	Ê	Ê	латинская заглавная буква Е с циркумфлексом
Ë	Ë	Ë	латинская заглавная буква Е с тремой
Ì	Ì	Ì	латинская заглавная буква I с тупым ударением
Í	Í	Í	латинская заглавная буква I с острым ударением

Таблица П1. 1 (продолжение)

Символ	Код	Вид	Описание
Î	Î	Î	латинская заглавная буква I с циркумфлексом
Ï	Ï	Ï	латинская заглавная буква I с тремой
Ð	Ð	D	латинские заглавные символы ETH
Ñ	Ñ	Ñ	латинская заглавная буква N с тильдой
Ò	Ò	6	латинская заглавная буква О с тупым ударением
Ó	Ó	6	латинская заглавная буква О с острым ударением
Ô	Ô	6	латинская заглавная буква 0 с циркумфлексом
Õ	Õ	б	латинская заглавная буква 0 с тильдой
Ö	Ö	Ö	латинская заглавная буква 0 с тремой
×	×	×	знак умножения
Ø	Ø	0	латинская заглавная буква 0 со штрихом
Ù	Ù	Ù	латинская заглавная буква U с тупым ударением
Ú	Ú	Ú	латинская заглавная буква U с острым ударением
Û	Û	Û	латинская заглавная буква U с циркумфлексом
Ü	Ü	Ü	латинская заглавная буква U с тремой
Ý	Ý	Ý	латинская заглавная буква Y с острым ударением
Þ	Þ	Þ	латинская заглавная буква THORN
à	à	à	латинская строчная буква А с тупым ударением
á	&##225;	á	латинская строчная буква А с острым ударением
â	&##226;	â	латинская строчная буква А с циркумфлексом
ã	ã	ã	латинская строчная буква А с тильдой
ä	ä	ä	латинская строчная буква А с тремой
å	å	å	латинская строчная буква А с верхним кружком
æ	æ	æ	латинские строчные буквы АЕ
ç	ç	ç	латинская строчная буква А с седилем

Таблица П1. 1 (продолжение)

Символ	Код	Вид	Описание
è	è	è	латинская строчная буква Е с тупым ударением
é	é	ё	латинская строчная буква Е с острым ударением
ê	ê	ё	латинская строчная буква Е с циркумфлексом
ë	ë	ё	латинская строчная буква Е *с* тремой
ì	ì	I	латинская строчная буква I с тупым ударением
í	í	í	латинская строчная буква I с острым ударением
î	î	î	латинская строчная буква I с циркумфлексом
ï	ï	ï	латинская строчная буква I с тремой
ð	ð	б	латинские строчные символы eth
ñ	ñ	ñ	латинская строчная буква N с тильдой
ò	ò	6	латинская строчная буква О с тупым ударением
ó	ó	6	латинская строчная буква 0 с острым ударением
ô	ô	ô	латинская строчная буква О с циркумфлексом
õ	õ	б	латинская строчная буква I с тильдой
ö	ö	б	латинская строчная буква I с тремой
÷	÷	÷	знак деления
ø	ø	0	латинская строчная буква О со штрихом
ù	ù	ù	латинская строчная буква U с тупым ударением
ú	ú	ú	латинская строчная буква U с острым ударением
û	û	û	латинская строчная буква U с циркумфлексом
ü	ü	0	латинская строчная буква U с тремой
ý	ý	ý	латинская строчная буква Y с острым ударением
þ	þ	þ	латинская строчная буква thorn
ÿ	ÿ	ÿ	латинская строчная буква Y с тремой
ƒ	ƒ	*f*	знак функции

Таблица П1. 1 (продолжение)

Символ	Код	Вид	Описание
			Греческие буквы
Α	Α	А	греческая заглавная буква альфа
Β	Β	В	греческая заглавная буква бета
Γ	Γ	Г	греческая заглавная буква гамма
Δ	Δ	Д	греческая заглавная буква дельта
Ε	Ε	Е	греческая заглавная буква эпсилон
Ζ	Ζ	Z	греческая заглавная буква дзета
Η	Η	II	греческая заглавная буква эта
Θ	Θ	0	греческая заглавная буква тета
Ι	Ι	I	греческая заглавная буква иота
Κ	Κ	К	греческая заглавная буква каппа
Λ	Λ	Л	греческая заглавная буква лямбда
Μ	Μ	Ml	греческая заглавная буква мю
Ν	Ν	N	греческая заглавная буква ню
Ξ	Ξ	Ξ	греческая заглавная буква кси
Ο	Ο	О	греческая заглавная буква омикрон
Π	Π	П	греческая заглавная буква пи
Ρ	Ρ	р	греческая заглавная буква ро
Σ	Σ	S	греческая заглавная буква сигма
Τ	Τ	Т	греческая заглавная буква тау
Υ	Υ	Y	греческая заглавная буква ипсилон
Φ	Φ	Φ	греческая заглавная буква фи
Χ	Χ	Х	греческая заглавная буква хи
Ψ	Ψ	Ψ	греческая заглавная буква пси
Ω	Ω	Ω	греческая заглавная буква омега
α	α	а	греческая строчная буква альфа
β	β	Р	греческая строчная буква бета
γ	γ	у	греческая строчная буква гамма
δ	δ	6	греческая строчная буква дельта
ε	ε	ε	греческая строчная буква эпсилон

Таблица П1. 1 (продолжение)

Символ	Код	Вид	Описание
ζ	ζ	ζ	греческая строчная буква дзета
η	η	η	греческая строчная буква эта
θ	θ	θ	греческая строчная буква тета
ι	ι	ι	греческая строчная буква иота
κ	κ	κ	греческая строчная буква каппа
λ	λ	λ	греческая строчная буква лямбда
μ	μ	μ	греческая строчная буква мю
ν	ν	ν	греческая строчная буква ню
ξ	ξ	ξ	греческая строчная буква кси
ο	ο	ο	греческая строчная буква омикрон
π	π	π	греческая строчная буква пи
ρ	ρ	ρ	греческая строчная буква ро
ς	ς	ς	греческая строчная буква сигма (final)
σ	σ	σ	греческая строчная буква сигма
τ	τ	τ	греческая строчная буква тау
υ	υ	υ	греческая строчная буква ипсилон
	φ	φ	греческая строчная буква фи
χ	χ	χ	греческая строчная буква хи
ψ	ψ	ψ	греческая строчная буква пси
ω	ω	ω	греческая строчная буква омега
			Стрелки
←	←	←	стрелка влево
↑	↑	↑	стрелка вверх
→	→	→	стрелка вправо
↓	↓	↓	стрелка вниз
↔	↔	↔	стрелка влево-вправо
Прочие символы			
♠	♠	♠	знак масти 'пики'
♣	♣	♣	знак масти 'трефы'
♥	♥	♥	знак масти 'червы'

Таблица П1.1 (окончание)

Символ	Код	Вид	Описание
♦	♦	4	знак масти 'бубны'
"	"	"	двойная кавычка
&	&	&	амперсанд
<	<	<	знак 'меньше'
>	>	>	знак 'больше'
ˆ	ˆ	ˆ	символ циркумфлекса (диакритический знак над гласной)
˜	˜	˜	тильда
™	™	™	знак торговой марки
		Знаки пунктуации	
•	•	•	маркер (bullet)
…	…	…	многоточие ...
‾	‾	‾	надчеркивание
⁄	⁄	/	косая дробная черта
		Общая пунктуация	
–	–	–	тире
—	—	—	длинное тире
‘	‘	‘	левая одиночная кавычка
’	’	’	правая одиночная кавычка
‚	‚	‚	нижняя одиночная кавычка
“	“	“	левая двойная кавычка
”	”	”	правая двойная кавычка
„	„	„	нижняя двойная кавычка

Приложение 2

Обозначения цветов в HTML

Таблица П2.1

Название	Шестнадцатеричное обозначение	Модель RGB
Aliceblue	#FOF8FF	240 248 255
Antiquewhite	#FAEBD7	250 235 215
Aqua	#00FFFF	0 255 255
Aquamarine	#7FFFD4	127 255 212
Azure	IFOFFFF	240 255 255
Beige	#F5F5DC	245 245 220
Bisque	#FFE4C4	255 228 196
Black	#000000	0 0 0
Blanchedalmond	#FFEBCD	255 235 205
Blue	#00FFFF	0 255 255
Blueviolet	#8A2BE2	138 43 226
Brown	#A52A2A	165 42 42
Burlywood	#DE8887	222 136 135
Cadetblue	#5F9EA0	95 158 160
Chocolate	ID2691E	210 105 30
Coral	#FF7F50	255 127 80
Cornflowerblue	#6495ED	100 149 237
Cornsilk	#FFF8DC	255 248 220
Crimson	#DC143C	220 20 60
Cyan	ttOOFFFF	0 255 255
Darkblue	#00008B	0 0 139
Darkcyan	#008B8B	0 139 139

Таблица **П2.1** *(продолжение)*

Название	Шестнадцатеричное обозначение	Модель RGB
Darkgoldenrod	#B8860B	184 134 11
Darkgray	#A9A9A9	169 169 169
Darkgreen	#006400	0 100 0
Darkkhaki	#BDB76D	189 183 109
Darkmagenta	#8B008B	139 0 139
Darkolivegreen	#556B2F	85 107 47
Darkorange	#FF8C00	255 140 0
Darkorchid	#9932CC	153 50 204
Darkred	#8B0000	139 0 0
Darksalmon	#E9967A	233 150 122
Darkseagreen	#8FBC8F	143 188 143
Darkslateblue	#483D8B	72 61 139
Darkslategray	#2F4F4F	47 79 79
Darkturquoise	#00CED1	0 206 209
Darkviolet	#9400D3	148 0 211
Deeppink	#FF1493	255 20 147
Deepskyblue	#00BFFF	0 191 255
Dimgray	#696969	105 105 105
Dodgerblue	#1E90FF	30 144 255
Firebrick	#B22222	178 34 34
Floralwhite	#FFFAF0	255 250 240
Forestgreen	#228B22	34 139 34
Fuchsia	#FF00FF	255 0 255
Gainsboro	#DCDCDC	220 220 220
Ghostwhite	#F8F8FF	248 248 255
Gold	#FFD700	255 215 0
Goldenrod	#DAA520	218 165 32
Gray	#808080	128 128 128
Green	#008000	0 128 0
Greenyellow	#ADFF2F	173 255 47
Honeydew	#F0FFF0	240 255 240

Таблица ***П2.1*** *(продолжение)*

Название	**Шестнадцатеричное** обозначение	Модель RGB
Hotpink	#FF69B4	255 105 180
Indianred	#CD5C5C	205 92 92
Indigo	#4B0082	75 0 130
Ivory	#FFFFFO	255 255 240
Khaki	#F0E68C	240 230 140
Lavender	#E6E6FA	230 230 250
Lavenderblush	#FFFOF5	255 240 245
Lemonchiffon	#FFFACD	255 250 205
Lightblue	#ADD8E6	173 216 230
Lightcoral	IF08080	240 128 128
Lightcyan	#EOFFFF	224 255 255
Lightgoldenrodyellow	#FAFAD2	250 250 210
Lightgreen	#90EE90	144 238 144
Lightpink	#FFB6C1	255 182 193
Lightsalmon	#FFA07A	255 160 122
Lightseagreen	#20B2AA	32 178 170
Lightskyblue	#87CEFA	135 206 250
Lightslategray	#778899	119 136 153
Lightsteelblue	#BOC4DE	176 196 222
Lightyellow	#FFFFE0	255 255 224
Lime	#00FF00	0 255 0
Limegreen	#32CD32	50 205 50
Linen	#FAFOF6	250 240 246
Magenta	#FFOOFF	255 0 255
Maroon	#800000	128 0 0
Mediumaquamarine	#66CDAA	102 205 170
Mediumblue	#0000CD	0 0 205
Mediumorchid	#BA55D3	186 85 211
Mediumpurple	#9370DB	147 112 219
Mediumseagreen	#3CB371	60 179 113
Mediumslateblue	#7B68EE	123 104 238

Таблица ***П2.1*** *(продолжение)*

Название	**Шестнадцатеричное** обозначение	**Модель** RGB
Mediumspringgreen	ttOOFA9A	0 250 154
Mediumturquoise	#48D1CC	72 209 204
Mediumvioletred	#C71585	199 21 133
Midnightblue	#191970	25 25 112
Mintcream	#F5FFFA	245 255 250
Mistyrose	#FFE4E1	255 228 225
Moccasin	#FFE4B5	255 228 181
Navajowhite	#FFDEAD	255 222 173
Navy	#000080	0 0 128
Oldlace	#FDF5E6	253 245 230
Olive	#808000	128 128 0
Olivedrab	#6B8E23	107 142 35
Orange	#FFA500	255 165 0
Orangered	#FF4500	255 69 0
Orchid	#DA70D6	218 112 214
Palegoldenrod	#EEE8AA	238 232 170
Palegreen	#98FB98	152 251 152
Paleturquoise	#AFEEEE	175 238 238
Palevioletred	#DB7093	219 112 147
Papayawhip	#FFEFD5	255 239 213
Peachpuff	#FFDAB9	255 218 185
Peru	#CD853F	205 133 63
Pink	#FFCOCB	255 192 203
Plum	#DDAODD	221 160 221
Powderblue	#BOEOE6	176 224 230
Purple	#800080	128 00 128
Red	#FF0000	255 0 0
Rosybrown	#BC8F8F	188 143 143
Royalblue	#4169E1	65 105 225
Salmon	#FA8072	250 128 114
Sandybrown	#F4A460	244 164 96

Таблица П2.1 (окончание)

Название	Шестнадцатеричное обозначение	Модель RGB
Seagreen	#2E2B57	43 46 87
Seashell	#FFE5EE	255 229 238
Sienna	#A0522D	160 82 45
Silver	#C0C0C0	192 192 192
Skyblue	#87CEEB	135 206 235
Slateblue	#6A5ACD	106 90 205
Slategray	#708090	112 128 144
Snow	#FFFAFA	255 250 250
Springgreen	#00FF7F	0 255 127
Steelblue	#4682B4	70 130 180
Tan	#D2B48C	210 180 140
Teal	#008080	0 128 128
Thistle	#D8BFD8	216 191 216
Tomato	#FF6347	255 99 71
Turquoise	#40E0D0	64 224 208
Violet	#EE82EE	238 130 238
Wheat	#F5DEB3	245 222 179
White	#FFFFFF	255 255 255
Whitesmoke	#F5F5F5	245 245 245
Yellow	#FFFF00	255 255 0
Yellowgreen	#9ACD32	154 205 50

Приложение 3

Полезные ссылки

Ссылки на русскоязычные и зарубежные интернет-ресурсы, посвященные языку разметки HTML, CSS и Dynamic HTML.

Русскоязычные ресурсы:

- Заметки HTML-кодера — http://htmlcoder.visions.ru/
- Спецификация HTML 4.0 (русский перевод официальной версии Консорциума W3C) — **http://www.stack.ru/~julia/HTML4/cover.phtml**
- CIT Forum (раздел HTML) — http://www.citforum.ru/internet/html/
- GetInfo.Ru — Компьютерная библиотека — **http://www.getinfo.ru/**
- HTML-справочник — http://html.manual.ru/
- HTML Book — http://www.htmlbook.ru/
- Report — сообщество экспертов (HTML) — **http://html.report.ru/**

Зарубежные ресурсы:

- HTML Goodies — http://www.htmlgoodies.com/
- HTMLHelp — http://www.htmlhelp.com/
- PageResource — http://www.pageresource.com/
- The Unicode Consortium and The Unicode Standard — **http://www.unicode.org/**
- W3Schools — http://www.w3schools.com/
- WebDeveloper — http://www.webdeveloper.com/html/
- Web Developer's Virtual Library — **http://www.wdvl.com/Authoring/**
- World Wide Web Consortium — **http://www.w3.org/**

Предметный указатель

K

L

M

N

O

P

R

S

T

U

V

W

X

А

Б

В

SoftLine direct

КАТАЛОГ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



- *Если вы хотите быть в курсе всех последних событий на рынке программного обеспечения,*
- *Если вы хотите получать наиболее полную информацию о программных продуктах из первых рук - от самих производителей,*
- *Если вы ведете честный бизнес и покупаете лицензионное ПО*

119991 г. Москва,
ул. Губкина, 8
(095) 232-0023
info@softline.ru
www.softline.ru

ЗНАЧИТ ВАША ЖИЗНЬ МОЖЕТ СТАТЬ ПРОЩЕ!

Подпишитесь на новый полноцветный каталог, издаваемый одним из крупнейших поставщиков программного обеспечения в России, и вы будете регулярно получать его по почте. Кроме того, по вашему желанию на ваш электронный адрес будут регулярно приходить еженедельные новости рынка программного обеспечения от компании SoftLine.

Подписка **БЕСПЛАТНО**
для руководителей и специалистов
в области информационных технологий,
представляющих организации,
имеющие более 10 компьютеров!



Книги издательства "БХВ-Петербург" в продаже:

Серия "В подлиннике"

Андреев А. и др. MS Windows XP: Home Edition и Professional	848 с.
Андреев А. и др. Windows 2000 Professional. Русская версия	700 с.
Андреев А. и др. Microsoft Windows 2000 Server. Русская версия	960 с.
Андреев А. и др. Новые технологии Windows 2000	576 с.
Андреев А. и др. Microsoft Windows 2000 Server и Professional. Русские версии	1056 с.
Ахаян Р. Macromedia ColdFusion	672 с.
Браун М. HTML 3.2 (с компакт-диском)	1040 с.
Вебер Дж. Технология Java (с компакт-диском)	1104 с.
Власенко С. Компакт-диск с примерами к книгам серии "В подлиннике": "MS Office XP *в* целом", "MS Access 2002", "MS Word 2002", "MS Excel 2002"	32 с.
Власенко С. Microsoft Word 2002	992 с.
Гофман В., Хомоненко А. Delphi 6	1152 с.
Долженков В. MS Excel 2002	1072 с
Закер К. Компьютерные сети. Модернизация и поиск неисправностей	1008 с.
Колесниченко О., Шишигин И. Аппаратные средства PC, 4-е издание	1024 с.
Мамаев Е. MS SQL Server 2000	1280 с.
Матросов А. и др. HTML 4.0	672 с.
Михеева В., Харитонова И. Microsoft Access 2000	1088 с.
Михеева В., Харитонова И. Microsoft Access 2002	1040 с.
Новиков Ф., Яценко А. Microsoft Office 2000 в целом	728 с.
Новиков Ф., Яценко А. Microsoft Office XP в целом	928 с.
Ноутон П., Шилдт Г. Java 2	1072с.
Пауэлл Т. Web-дизайн	1024 с.
Персон Р. Word 97	1120 с.
Питц М., Кирк Ч. XML	736 с.
Пономаренко С. Adobe Illustrator 9.0	608 с.
Пономаренко С. Adobe Photoshop 6.0	832 с.
Пономаренко С. CorelDRAW 9	576 с.
Пономаренко С. Macromedia FreeHand 9	432 с.
Русеев С. WAP: технология и приложения	432 с.
Секунов Н. Обработка звука на PC (с дискетой)	1248 с.
Сузи Р. Python (с компакт-диском)	768 с.
Тайц А. М., Тайц А. А. Adobe PageMaker 7.0	784 с.
Тайц А. М., Тайц А. А. Adobe InDesign	704 с.
Тайц А. М., Тайц А. А. CorelDRAW 9: все программы пакета	1136 с.
Тайц А. М., Тайц А. А. CorelDRAW 10: все программы пакета	1136 с.
Тихомиров Ю. Microsoft SQL Server 7.0	720 с.

Уильяме Э. и др. Active Server Pages (с компакт-диском)	672 с.
Усаров Г. Microsoft Outlook 2002	656 с.
Ханкт Ш. Эффекты CorelDRAW (с компакт-диском)	704 с.

Серия *"Мастер"*

CD-ROM с примерами к книгам "Ресурсы MS Windows NT Server 4.0" и "Сетевые средства Windows NT Server 4"	
Microsoft Press. Электронная коммерция. B2B-программирование (с компакт-диском)	368 с.
Microsoft Press. Visual Basic 6.0	992 с.
Microsoft Press. Ресурсы MS Windows NT Server 4.0	752 с.
Айзеке С. Dynamic HTML (с компакт-диском)	496 с.
Анин Б. Защита компьютерной информации	384 с.
Асбари С. Корпоративные решения на базе Linux	496 с.
Березин С. Факс-модемы: выбор, подключение, выход в Интернет	256 с.
Березин С. Факсимильная связь в Windows	250 с.
Борн Г. Реестр Windows 98 (с дискетой)	496 с.
Бухвалов А. и др. Финансовые вычисления для профессионалов	320 с.
Валиков А. Технология XSLT	432 с.
Габбасов Ю. Internet 2000	448 с.
Гарбар П. Novell GroupWise 5.5: система электронной почты и коллективной работы	480 с.
Гарнаев А. Microsoft Excel 2000: разработка приложений	576 с.
Гарнаев А. Excel, VBA, Internet в экономике и финансах	816 с.
Гарнаев А., Гарнаев С. Web-программирование на Java и JavaScript	1040 с.
Гордеев О. Программирование звука в Windows (с дискетой)	384 с.
Гофман В., Хомоненко А. Работа с базами данных в Delphi	656 с.
Дарахвелидзе П. и др. Программирование в Delphi 5 (с дискетой)	784 с.
Дронов В. JavaScript в Web-дизайне	880 с.
Дубина А. и др. MS Excel в электронике и электротехнике	304 с.
Дубина А. Машиностроительные расчеты в среде Excel 97/2000 (с дискетой)	416 с.
Дунаев С. Технологии Интернет-программирования	480 с.
Жарков С. Shareware: профессиональная разработка и продвижение программ	320 с.
Зима В. и др. Безопасность глобальных сетевых технологий	320 с.
Киммел П. Borland C++ 5	976 с.
Костарев А. PHP в Web-дизайне	592 с.
Краснов М. DirectX. Графика в проектах Delphi (с компакт-диском)	416 с.
Краснов М. Open GL в проектах Delphi (с дискетой)	352 с.
Кубенский А. Создание и обработка структур данных в примерах на Java	336 с.
Кулагин Б. 3ds max 4: от объекта до анимации	448 с.
Купенштейн В. MS Office и Project в управлении и делопроизводстве	400 с.
Куприянов М. и др. Коммуникационные контроллеры фирмы Motorola	560 с.
Лавров С. Программирование. Математические основы, средства, теория	304 с.
Лукацкий А. Обнаружение атак	624 с.

Матросов А. Maple 6. Решение задач высшей математики и механики	528 с.
Медведев Е., Трусова В. "Живая" музыка на PC (с дискетой)	720 с.
Мешков А., Тихомиров Ю. Visual C++ и MFC, 2-е издание (с дискетой)	1040 с.
Миронов Д. Создание Web-страниц в MS Office 2000	320 с.
Мещеряков Е., Хомоненко А. Публикация баз данных в Интернете	560 с.
Михеева В., Харитонова И. Microsoft Access 2000: разработка приложений	832 с.
Новиков Ф. и др. Microsoft Office 2000: разработка приложений	680 с.
Нортон П. Разработка приложений в Access 97 (с компакт-диском)	656 с.
Одинцов И. Профессиональное программирование. Системный подход	512 с.
Олифер В., Олифер Н. Новые технологии и оборудование IP-сетей	512 с.
Подольский С. и др. Разработка интернет-приложений в Delphi (с дискетой)	432 с.
Полещук Н. Visual LISP и секреты адаптации AutoCAD	576 с.
Понамарев В. COM и ActiveX в Delphi	320 с.
Пономаренко С. Adobe InDesign: дизайн и верстка	544 с.
Попов А. Командные файлы и сценарии Windows Scripting Host	320 с.
Приписнов Д. Моделирование в 3D Studio MAX 3.0 (с компакт-диском)	352 с.
Роббинс Дж. Отладка приложений	512с.
Рудометов В., Рудометов Е. PC: настройка, оптимизация и разгон, 2-е издание	336 с.
Русеев Д. Технологии беспроводного доступа. Справочник	352 с.
Соколенко П. Программирование SVGA-графики для IBM	432 с.
Тайц А. Каталог Photoshop Plug-Ins	464 с.
Тихомиров Ю. MS SQL Server 2000: разработка приложений	368 с.
Тихомиров Ю. SQL Server 7.0: разработка приложений	370 с.
Тихомиров Ю. Программирование трехмерной графики в Visual C++ (с дискетой)	256 с.
Трельсен Э. Модель COM и библиотека ATL 3.0 (с дискетой)	928 с.
Федоров А., Елманова Н. ADO в Delphi (с компакт-диском)	816 с.
Федорчук А. Офис, графика, Web в Linux	416 с.
Чекмарев А. Windows 2000 Active Directory	400 с.
Чекмарев А. Средства проектирования на Java *(с* компакт-диском)	400 с.
Шапошников И. Web-сайт своими руками	224 с.
Шапошников И. Интернет-программирование	224 с.
Шапошников И. Справочник Web-мастера. XML	304 с.
Шилдт Г. Теория и практика C++	416 с.
Яцюк О., Романычева Э. Компьютерные технологии в дизайне. Логотипы, упаковка, буклеты (с компакт-диском)	464 с.

Серия "Изучаем вместе с BHV"

Березин С. Internet у вас дома, 2-е издание	752 с.
Тайц А. Adobe Photoshop 5.0 (с дискетой)	448 с.

Серия "Самоучитель"

Ананьев А., Федоров А. Самоучитель Visual Basic 6.0	624 с.
Васильев В. Основы работы на ПК	448 с.
Гарнаев А. Самоучитель VBA	512 с.
Герасевич В. Самоучитель. Компьютер для врача	640 с.
Дмитриева М. Самоучитель JavaScript	512 с.
Долженков В. Самоучитель Excel 2000 (с дискетой)	368 с.
Исагулиев К. Macromedia Dreamweaver 4	560 с.
Исагулиев К. Macromedia Flash 5	368 с.
Кетков Ю., Кетков А. Практика программирования: Бейсик, Си, Паскаль (с дискетой)	480 с.
Кирьянов Д. Самоучитель Adobe Premiere 6.0	432 с.
Кирьянов Д. Самоучитель MathCAD 2001	544 с.
Коркин И. Самоучитель Microsoft Internet Explorer 6.0	288 с.
Котеров Д. Самоучитель PHP 4	576 с.
Культин Н. Программирование на Object Pascal в Delphi 6 (с дискетой)	528 с.
Культин Н. Самоучитель. Программирование в Turbo Pascal 7.0 и Delphi, 2-е издание (с дискетой)	416 с.
Леоненков А. Самоучитель UML	304 с.
Матросов А., Чаунин М. Самоучитель Perl	432 с.
Омельченко Л., Федоров А. Самоучитель Microsoft FrontPage 2002	576 с.
Омельченко Л., Федоров А. Самоучитель Windows 2000 Professional	528 с.
Омельченко Л., Федоров А. Самоучитель Windows Millennium	464 с.
Пекарев Л. Самоучитель 3D Studio MAX 4.0	370 с.
Полещук Н. Самоучитель AutoCad 2000 и Visual LISP, 2-е издание	672 с.
Полещук Н. Самоучитель AutoCAD 2002	608 с.
Понамарев В. Самоучитель Kylix	416 с.
Секунов Н. Самоучитель Visual C++ 6 (с дискетой)	960 с.
Секунов Н. Самоучитель C#	576 с.
Сироткин С. Самоучитель WML и WMLScript	240 с.
Тайц А. М., Тайц А. А. Самоучитель Adobe Photoshop 6 (с дискетой)	608 с.
Тайц А. М., Тайц А. А. Самоучитель CorelDRAW 10	640 с.
Тихомиров Ю. Самоучитель MFC (с дискетой)	640 с.
Хабибуллин И. Самоучитель Java	464 с.
Хомоненко А. Самоучитель Microsoft Word 2002	624 с.
Шапошников И. Интернет. Быстрый старт	272 с.
Шапошников И. Самоучитель HTML 4	288 с.
Шилдт Г. Самоучитель C++, 3-е издание (с дискетой)	512 с.

Серия "Компьютер и творчество"

Деревских В. Музыка на PC своими руками	352 с.
Дунаев В. Сам себе Web-дизайнер	512 с.
Дунаев В. Сам себе Web-мастер	288 с.

Людиновсков С. Музыкальный видеоклип своими руками	320 с.
Петелин Р., Петелин Ю. Аранжировка музыки на PC	272 с.
Петелин Р., Петелин Ю. Звуковая студия в PC	256 с.
Петелин Р., Петелин Ю. Музыка на PC. Cakewalk Pro Audio 9. Секреты мастерства	420 с.
Петелин Р., Петелин Ю. Музыка на PC. Cakewalk. "Примочки" и плагины	272 с.
Петелин Р., Петелин Ю. Музыкальный компьютер. Секреты мастерства	608 с.
Петелин Р., Петелин Ю. Персональный оркестр в PC	240 с.

Серия "Учебное пособие"

Бенькович Е. Практическое моделирование динамических систем (с компакт-диском)	464 с.
Гомоюнов К. Транзисторные цепи	240 с.
Дорот В. Толковый словарь современной компьютерной лексики, 2-е издание	512 с.
Культин Н. C/C++ *в* задачах и примерах	288 с.
Культин Н. Turbo Pascal в задачах и примерах	256 с.
Порев В. Компьютерная графика	432 с.
Робачевский Г. Операционная система Unix	528 с.
Сафронов И. Бейсик в задачах и примерах	224 с.
Солонина А. и др. Алгоритмы и процессоры цифровой обработки сигналов	464 с.
Солонина А. и др. Цифровые процессоры обработки сигналов фирмы MOTOROLA	512 с.
Угрюмов Е. Цифровая схемотехника	528 с.
Шелест В. Программирование	592 с.

Серия "Знакомьтесь"

Надеждин Н. Карманные компьютеры	304 с.
Надеждин Н. Портативные компьютеры	288 с.
Надеждин Н. Знакомьтесь, цифровые фотоаппараты	304 с.

Серия "Быстрый старт"

Васильева В. Персональный компьютер. Быстрый старт	480 с.
Гофман В., Хомоненко А. Delphi. Быстрый старт	288 с.
Дмитриева М. JavaScript. Быстрый старт	336 с.
Культин Н. Microsoft Excel. Быстрый старт	208 с.
Хомоненко А., Гридин. В. Microsoft Access. Быстрый старт	304 с.

Азбука программирования в HTML

bhv

Алексей Петюшкин, автор более 150 информационных, аналитических и научных статей в различных компьютерных и образовательных изданиях ("CHIP-Россия", "ComputerPrice", "Магия ПК", "Компьютерная Россия", "Мир Интернет", "Компьютерная Газета", "Новое Знание" и пр.), автор книги "Основы баннерной рекламы" ("БХВ-Петербург", 2002 г.). Опыт работы в интернет-индустрии свыше 7 лет.

HTML

ЭКСПРЕСС•КУРС

Язык гипертекстовой разметки HTML постоянно набирает популярность, причем не только в сфере интернет-технологий, но и в области презентационных услуг, рекламно-выставочной деятельности, в разработке программного обеспечения.

Возможность быстрого и эффективного освоения языка HTML дает предлагаемый экспресс-курс, он будет интересен и как учебник для начинающих, и как справочник для более опытных пользователей. Для этого совсем необязательно быть профессиональным программистом, достаточно по шагам пройти вместе с автором весь курс и вы сможете грамотно и профессионально создавать привлекательные и интересные электронные документы.

ISBN 5-94157-317-0

9 785941 573172

интернет—магазин
www.computerbook.ru